

TraCS: Trajectory Collection in Continuous Space under Local Differential Privacy

Ye Zheng

Rochester Institute of Technology
Rochester, NY, USA

Yidan Hu

Rochester Institute of Technology
Rochester, NY, USA

Abstract

Trajectory collection is essential for location-based services, yet it can reveal highly sensitive information about users, such as daily routines and activities, raising serious privacy concerns. Local Differential Privacy (LDP) offers strong privacy guarantees for users even when the data collector is untrusted. However, existing trajectory collection methods under LDP are largely confined to discrete location spaces, where the size of the location space affects both privacy guarantees and trajectory utility. Moreover, many real-world applications—such as flying trajectories or wearable-sensor traces—naturally operate in continuous spaces, making these discrete-space methods inadequate.

This paper shifts the focus from discrete to continuous spaces for trajectory collection under LDP. We propose two methods: TraCS-D, which perturbs the direction and distance of locations, and TraCS-C, which perturbs the Cartesian coordinates of locations. Both methods are theoretically and experimentally analyzed for trajectory utility in continuous spaces. TraCS can also be applied to discrete spaces by rounding perturbed locations to any discrete space embedded in the continuous space. In this case, the privacy and utility guarantees of TraCS are independent of the number of locations in the space, and each perturbation requires only $\Theta(1)$ time complexity. Evaluation results on discrete location spaces validate the efficiency advantage and demonstrate that TraCS outperforms state-of-the-art methods with improved trajectory utility, particularly for large privacy parameters.

Keywords

local differential privacy, continuous space, trajectory data

1 Introduction

Trajectory data from users—sequences of locations describing movement over time—are a fundamental resource for activity analysis and location-based services, including activity classification and routine detection. However, directly collecting trajectory data raises significant privacy concerns, especially given users’ growing privacy expectations and stringent regulations [1, 22]. Anonymization techniques like k -anonymity [23, 33] have proven insufficient to protect trajectory privacy [7]. Thus, stronger privacy guarantees are necessary for such data.

Local differential privacy (LDP) provides a provable worst-case privacy guarantee by ensuring the hardness of distinguishing a

user’s data from any released data. The privacy level of an LDP mechanism is quantified by its privacy parameter ϵ , which also generally implies the extent of data perturbation. Through LDP, users can release their perturbed trajectory data to an untrusted data collector while still benefiting from the services provided by the collector.

Existing LDP methods for trajectory collection are designed for discrete spaces. They leverage discrete LDP mechanisms, such as the Exponential mechanism [31], to perturb the trajectory data. Discrete LDP mechanisms are explicitly defined on the size of the location space. As a result, these methods either rely on properly splitting the entire location space into grids [35] or assume that the location space consists of a finite set of labeled locations (i.e. points of interest) [10, 44]. However, relying on discrete location spaces has three inherent limitations: (i) Their privacy guarantees are inherently tied to the discrete set. For example, a space with 10 locations offers weaker effective privacy than one with 100 locations: even a trivial inference strategy that always outputs a fixed location succeeds with probability at least $1/10$ in the former, regardless of the privacy parameter ϵ .^{*} (ii) Their efficacy and efficiency are often limited by the domain size. As the number of candidate locations grows, the probability of a mechanism outputting the true location diminishes. Furthermore, the widely used Exponential mechanism incurs linear sampling complexity in the domain size, making the generation of each perturbed location computationally expensive.[†] (iii) Discrete methods are not directly applicable to inherently continuous location spaces, such as those arising from flying and sailing trajectories or sensor data from wearable devices. Although one can discretize a continuous space before applying discrete methods, this inherits the aforementioned limitations. Moreover, choosing a suitable discretization granularity that balances privacy, utility, and efficiency is non-trivial.[‡]

To address these limitations, this paper shifts the focus from discrete to continuous spaces for trajectory collection under LDP. A continuous space represents locations as real-valued points, such as GPS coordinates in $[-180, 180] \times [-90, 90]$, and thus contains infinitely many candidate locations. Collecting trajectory data directly in continuous spaces is natural for many applications and offers three key advantages. (i) The privacy guarantee does not depend on the cardinality of a discretized location set (ii) The sampling mechanism operates directly on the continuous space, so its efficacy and efficiency are not constrained by the space size. (iii) Perturbed locations can be post-processed (e.g. rounded) to any

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license visit <https://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Proceedings on Privacy Enhancing Technologies 2026(3), 118–138

© 2026 Copyright held by the owner/author(s).

<https://doi.org/10.56553/popets-2026-0074>



^{*}Appendix C.1 provides details on the space-dependence of indistinguishability.

[†]Appendix C.2 details the Exponential mechanism’s efficiency and data utility (efficacy) limitations.

[‡]Appendix C.3 further discusses the challenges of adapting discrete mechanisms to continuous spaces.

discrete space embedded within the continuous space, making the approach applicable to both continuous and discrete settings.

Designing LDP trajectory collection methods for continuous spaces poses unique challenges: they must guarantee LDP for every location in the continuous domain; they must preserve sufficient trajectory utility for downstream tasks relative to the original trajectories (formalized in Section 4); and they must generate perturbed locations efficiently to support real-time applications.

This paper proposes two LDP methods for trajectory collection in continuous spaces. The main insight is to decompose the 2D continuous space into 1D subspaces and design mechanisms for each subspace. Our mechanisms build on existing utility-optimized piecewise-based mechanisms [47] for 1D bounded numerical domains.[§] Based on two different decompositions of the continuous space, we propose two methods: TraCS-D and TraCS-C. The method TraCS-D treats the continuous space as the composition of a direction space and a distance space. We exploit the direction information of trajectories and design a direction perturbation mechanism to ensure LDP for the direction space. Combined with a piecewise-based mechanism for the distance space, TraCS-D ensures LDP for the continuous 2D space. The method TraCS-C treats the continuous space as the Cartesian product of two distance spaces. Each location is represented by a 2D Cartesian coordinate, and TraCS-C perturbs these coordinates. Both methods guarantee LDP for the entire 2D continuous space. They can also be applied to any discrete space embedded in the continuous domain by rounding perturbed locations to their nearest discrete points. Compared with existing methods based on the Exponential mechanism [10, 44], TraCS is unaffected by the number of locations in the location space and require only $\Theta(1)$ time to generate each perturbed location. Specifically, our contributions are as follows:

- To the best of our knowledge, this is the first work to develop trajectory collection methods for continuous spaces under pure LDP. We highlight the benefits of operating directly in continuous spaces (rather than discretizing) and propose TraCS-D and TraCS-C accordingly. Our key insight is to decompose the 2D continuous space into two 1D subspaces and to build 2D trajectory perturbation mechanisms from existing utility-optimized 1D piecewise-based mechanisms, leveraging the direction and coordinate information in continuous trajectories. We theoretically and experimentally analyze their trajectory utility.
- Our method TraCS also applies to discrete spaces. Compared with existing approaches for discrete spaces, TraCS has substantially lower computational complexity for generating perturbed locations. Experiments on discrete location spaces validate this efficiency, with an average runtime below 1% of that of NGram, L-SRR, and ATP [10, 35, 44], three state-of-the-art LDP methods designed for discrete spaces. In addition, the results show that TraCS generally achieves better trajectory utility, particularly at larger privacy parameters.

[§]Another category of LDP mechanisms for bounded numerical domains is truncated mechanisms (e.g. the truncated Laplace mechanism [17, 26]). While such mechanisms can be incorporated into TraCS, they are more complex and typically less effective than piecewise-based mechanisms. Section 3.5.6 provides details, and Section 4 includes experimental comparisons.

2 Preliminaries

This section formulates the problem, introduces LDP, and reviews three trajectory collection methods under LDP. We highlight their limitations in continuous spaces, which motivates the need for new LDP methods for trajectory collection in such settings.

2.1 Problem Formulation

A typical trajectory collection schema consists of a set of trajectories from users and one collector. Each trajectory $\mathcal{T}$ is a sequence of locations $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$, where $\tau_i \in \mathcal{S}$ and $\mathcal{S} \subset \mathbb{R}^2$ is a continuous and bounded domain. The collector needs to collect these trajectories to provide analysis or location-based services.

However, the collector is untrusted and may act as an adversary attempting to infer users' sensitive data. Therefore, releasing the sensitive trajectories to the collector poses a privacy risk. To protect privacy, users perturb their trajectories using a privacy mechanism $\mathcal{M} : \mathcal{S} \rightarrow \text{Range}(\mathcal{M})$, and then send the perturbed trajectories $\mathcal{T}' = \{\tau'_1, \tau'_2, \dots, \tau'_n\}$ to the collector.

We aim to design a mechanism $\mathcal{M}$ that satisfies ϵ -LDP, providing provable privacy guarantee for sensitive trajectories. Additionally, $\mathcal{M}$ should preserve utility, ensuring perturbed trajectories remain comparable to the originals, while incurring only modest computational overhead suitable for real-time applications.

2.2 Local Differential Privacy

DEFINITION 1 (ϵ -LDP [12]). A perturbation mechanism $\mathcal{M} : \mathcal{X} \rightarrow \text{Range}(\mathcal{M})$ satisfies ϵ -LDP, if for two arbitrary input x_1 and x_2 , the probability ratio of outputting the same y is bounded:

$$\forall x_1, x_2 \in \mathcal{X}, \forall y \in \text{Range}(\mathcal{M}) : \frac{\Pr[\mathcal{M}(x_1) = y]}{\Pr[\mathcal{M}(x_2) = y]} \leq \exp(\epsilon).$$

If $\mathcal{M}(x)$ is continuous, the probability $\Pr[\cdot]$ is replaced by probability density function (*pdf*). Intuitively, Definition 1 represents the difficulty of distinguishing x_1 and x_2 given y . Lower values of the privacy parameter $\epsilon \in [0, +\infty)$ mean higher privacy. For example, $\epsilon = 0$ requires $\mathcal{M}$ to map two arbitrary inputs to any output y with the same probability, thus the perturbed output contains no distribution information of the sensitive input, making any hypothesis-testing method to infer the sensitive x powerless.

THEOREM 1 (SEQUENTIAL COMPOSITION OF LDP [15, 44]). Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two mechanisms that satisfy ϵ_1 and ϵ_2 -LDP, respectively. Their composition, defined as $\mathcal{M}_{1,2} := (\mathcal{M}_1, \mathcal{M}_2) : (\mathcal{X}_1, \mathcal{X}_2) \rightarrow (\text{Range}(\mathcal{M}_1), \text{Range}(\mathcal{M}_2))$, satisfies $(\epsilon_1 + \epsilon_2)$ -LDP.

Sequential composition enables (i) designing LDP mechanisms for 2D data by composing two 1D mechanisms, and (ii) perturbing multiple locations in a trajectory (i.e. repeated composition) while still providing an overall LDP guarantee.

2.2.1 Piecewise-based Mechanism. When the input domain $\mathcal{X}$ is continuous and bounded, state-of-the-art LDP mechanisms are often *piecewise-based* [25, 37, 47]. They generate an output by sampling from a probability distribution over $\mathcal{X}$ whose density is defined in a piecewise manner. Below, we present a unified formulation that captures this family of mechanisms.

DEFINITION 2 (PIECEWISE-BASED MECHANISM). A piecewise-based mechanism $\mathcal{M} : \mathcal{X} \rightarrow \text{Range}(\mathcal{M})$ is a family of probability

distributions that, given input $x \in \mathcal{X}$, outputs $y \in \text{Range}(\mathcal{M})$ according to

$$pdf[\mathcal{M}(x) = y] = \begin{cases} p_\varepsilon & \text{if } y \in [l_{x,\varepsilon}, r_{x,\varepsilon}), \\ p_\varepsilon / \exp(\varepsilon) & \text{otherwise,} \end{cases}$$

where p_ε is the sampling probability w.r.t. ε , $[l_{x,\varepsilon}, r_{x,\varepsilon})$ is the sampling interval w.r.t. x and ε . $\text{Range}(\mathcal{M}) \supseteq \mathcal{X}$ is also a continuous and bounded domain.

Piecewise-based mechanisms sample an output y with a high probability p_ε from the interval $[l_{x,\varepsilon}, r_{x,\varepsilon})$, and with a lower probability from the remaining two pieces, while still satisfying ε -LDP. Representative instantiations include OGPM [47] and PM [37] for mean estimation, and SW [25] for distribution estimation. Our perturbation mechanisms build on the 1D piecewise-based mechanism in OGPM to design new 2D mechanisms tailored to trajectory collection.

2.2.2 k -RR and Exponential Mechanism. If $\mathcal{X}$ is discrete, especially when $|\mathcal{X}|$ is not large [38], a classical LDP mechanism is k -RR (or GRR in some literature) [21].

DEFINITION 3. k -RR is a sampling mechanism $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{X}$ that, given $|\mathcal{X}| = k$ and input $x \in \mathcal{X}$, outputs $y \in \mathcal{X}$ according to

$$\Pr[\mathcal{M}(x) = y] = \begin{cases} \frac{\exp(\varepsilon)}{|\mathcal{X}| - 1 + \exp(\varepsilon)} & \text{if } y = x, \\ \frac{1}{|\mathcal{X}| - 1 + \exp(\varepsilon)} & \text{otherwise.} \end{cases}$$

k -RR outputs the truth x with a higher probability or outputs other values with a lower probability, while satisfying ε -LDP.

When there is a semantic distance (score) function between elements in $\mathcal{X}$, the Exponential mechanism [31] is more widely used. Unlike k -RR, which treats other values (except x) equally, the Exponential mechanism leverages a score function to assign different probabilities. k -RR is a special case of the Exponential mechanism when the score function is a binary indicator function.

DEFINITION 4 (EXPONENTIAL MECHANISM [31]). Given a score function $d : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, a privacy parameter $\varepsilon > 0$, and a set of possible outputs $\mathcal{Y}$, the Exponential mechanism $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Y}$ is defined by:

$$\Pr[\mathcal{M}(x) = y] = \frac{\exp\left(\frac{\varepsilon d(x, y)}{2\Delta d}\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\frac{\varepsilon d(x, y')}{2\Delta d}\right)},$$

where $\Delta d = \max_{x, y, y' \in \mathcal{Y}} |d(x, y) - d(x, y')|$ is the sensitivity of the score function d .

In the context of trajectory collection, common score functions include the negative Euclidean distance $d(x, y) := -\|x - y\|_2$, great circle distance [44], etc. Under such score functions (e.g. $d(x, y) = -\|x - y\|_2$), locations closer to the true location x have larger scores $d(x, y)$ and thus receive higher sampling probabilities, making them more likely to be selected.

2.2.3 Privacy Parameter for Trajectory Data. When applying LDP to trajectory data, there are two strategies for setting the privacy parameter. (i) Treating each location τ_i in a trajectory $\mathcal{T}$ as a data item and applying an ε -LDP mechanism to perturb each location;

Table 1: Comparison with existing methods.

	Main technique	LDP guarantee
NGram [10]	Hierarchical decomposition	Discrete space
L-SRR [35]	Staircase RR mechanism	
ATP [44]	Direction perturbation	
This paper (TraCS)	Direction ^a & coordinate perturbation	Continuous space

^a Designed for continuous direction space (different from ATP).

(ii) Treating the entire trajectory $\mathcal{T}$ as a data item and applying an ε -LDP mechanism to perturb the whole trajectory.[†] The first strategy is cleaner and convertible to the second strategy via the Sequential Composition Theorem 1, whereas the second depends on the trajectory length, which complicates the analysis and the design of mechanisms under a fixed ε . Therefore, we primarily adopt the first strategy in this paper, and include experiments for the second in Appendix C.10 for completeness.

2.3 Existing Methods

Table 1 summarizes three state-of-the-art trajectory collection methods under pure LDP, along with our approach. Other works that emphasize external knowledge rather than new perturbation mechanisms are discussed in Appendix A.

NGram [10] hierarchically decomposes the physical space $\mathcal{S}$ into fine-grained discrete regions with semantic labels. Concretely, $\mathcal{S}$ is first partitioned into small spatial regions R_s , then further refined into category-specific regions R_c , and finally augmented with temporal information R_t . A resulting “gram” can be written as $r_{set} = \{\text{mountain, church, 3am}\}$. This decomposition represents a trajectory $\mathcal{T}$ at the knowledge level, enabling the use of public knowledge to define *reachability*, i.e. to constrain the set of feasible perturbed trajectories. For instance, visiting a church at 3am is unlikely because it is typically closed, which reduces the reachable set of $\mathcal{T}'$. Finally, NGram applies the Exponential mechanism to sample a perturbed trajectory from this reachable set.

L-SRR [35] introduces the staircase randomized response (SRR) mechanism for location perturbation. Unlike k -RR, which assigns only two probabilities—one for the true location and another for all other locations—the SRR mechanism employs a staircase-shaped probability distribution. L-SRR recursively partitions the physical space $\mathcal{S}$ and forms location groups $G_1, G_2, \dots, G_g$ based on their distances to the true location. By integrating the SRR mechanism, L-SRR assigns higher probabilities to locations in groups that are closer to the true location, and lower probabilities to those farther away. This approach is similar to the Exponential mechanism, but uses group-level distance instead of location-level distance.

ATP [44] perturbs directions to restrict the set of feasible perturbed locations. For a trajectory $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$, when perturbing τ_{i+1} , ATP first perturbs the direction from τ_i to τ_{i+1} . Specifically, it partitions the direction space into k sectors and applies k -RR to

[†]This concerns how to set the privacy parameter for trajectory data, rather than defining the “trajectory space” as the input space in LDP mechanisms. We focus on LDP mechanisms for location spaces in this paper, as the trajectory space in $\mathbb{R}^2$ is infinite-dimensional.

sample a sector. The perturbed location τ'_{i+1} is then constrained to lie within the sampled sector, which reduces the candidate set of τ'_{i+1} ; ATP finally samples τ'_{i+1} from this constrained set using the Exponential mechanism.

We summarize their limitations in the context of continuous spaces as follows:

- Existing methods are designed for discrete location spaces. Discretizing a continuous domain is possible, but choosing an appropriate granularity is non-trivial.
- Beyond discretization, methods that rely on the Exponential mechanism (e.g. NGram and ATP) typically incur high computational cost for evaluating scores and sampling, and their utility can be sensitive to the spatial distribution of candidate locations. L-SRR requires constructing distance-based groups for every location, leading to similar scalability issues. Moreover, approaches that depend on public knowledge (e.g. NGram) are hard to design in continuous spaces (where the location set is infinite and semantics are difficult to define), are often dataset-specific, and the public knowledge is typically *known* to the adversary, which can further weaken privacy protection.

Appendix C.2 details the limitations of the Exponential mechanism. In brief, perturbing a location requires evaluating the score function $d(x, y)$ for all candidate outputs y associated with x , which takes $\Theta(m)$ time where m is the number of locations in the discrete space. Moreover, generating each perturbed location requires sampling from an m -piece cumulative distribution function (CDF), which also incurs $\Theta(m)$ time. Such a per-location cost is prohibitive for large-scale location spaces. Finally, because the sampling probabilities are induced by the score function, they can be sensitive to the spatial distribution of candidate locations and their relative distances, which may further degrade utility.

To address these limitations, we propose TraCS, a novel trajectory collection method that ensures rigorous LDP guarantees for trajectories in continuous spaces. TraCS is carefully designed to tackle the privacy, utility, and efficiency challenges inherent in handling continuous domains without relying on discretization.

3 Our Method: TraCS

This section presents our methods, TraCS-D and TraCS-C, along with their theoretical analyses and extensions. Notations used in this section can be found in Table 2.

3.1 Location Space

We consider the location space constrained by a pair of longitude and latitude lines, i.e. $\mathcal{S} \subset \mathbb{R}^2$ is a rectangular area $[a_{\text{sta}}, a_{\text{end}}] \times [b_{\text{sta}}, b_{\text{end}}]$, where $a_{\text{sta}}, a_{\text{end}}$ denote the longitudes and $b_{\text{sta}}, b_{\text{end}}$ denote the latitudes. Consequently, each location $\tau_i \in \mathcal{S}$ has a natural representation as a pair of coordinates (a_i, b_i) . This representation aligns with real-world location data (e.g. GPS data) and existing trajectory collection methods for discrete spaces [10, 11, 19, 35, 44].

The core idea behind having various LDP methods for the location space $\mathcal{S}$ stems from its multiple decompositions. One decomposition means $\mathcal{S}$ can be represented by several subspaces. For example, in addition to the aforementioned longitude-latitude coordinate representation, each location $\tau_i \in \mathcal{S}$ can also be represented

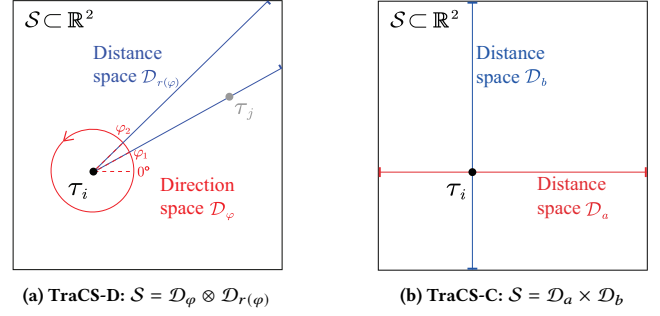


Figure 1: Two decompositions of location space $\mathcal{S}$ at τ_i . Any other location $\tau_j \in \mathcal{S}$ can be represented by direction-distance coordinates $(\varphi, r(\varphi))$ or Cartesian coordinates (a, b) .

Table 2: Notations used in this paper.

Notation	Meaning
$\mathcal{T} := \{\tau_1, \tau_2, \dots, \tau_n\}$	Sensitive trajectory
$\mathcal{T}' := \{\tau'_1, \tau'_2, \dots, \tau'_n\}$	Perturbed trajectory
$\mathcal{S} := [a_{\text{sta}}, a_{\text{end}}] \times [b_{\text{sta}}, b_{\text{end}}]$	Continuous location space
$\mathcal{M}_\phi$	Direction perturbation mechanism
$\mathcal{M}_r$	Distance perturbation mechanism
$\varphi \in \mathcal{D}_\varphi := [0, 2\pi)$	Direction & direction space
φ'	Perturbed direction
$r(\varphi) \in \mathcal{D}_{r(\varphi)}$	Distance & distance space along φ
$r'(\varphi)$	Perturbed distance along φ
$\bar{r}(\varphi) \in [0, 1)$	Normalized distance over $\mathcal{D}_{r(\varphi)}$

by a direction component and a distance component. Based on these decompositions, we propose two LDP methods: TraCS-D, which perturbs the direction and distance subspaces, and TraCS-C, which perturbs subspaces of the Cartesian coordinates.

3.2 TraCS-D

Continuous space $\mathcal{S}$ can be represented as the composition of a direction space $\mathcal{D}_\varphi$ and a distance space $\mathcal{D}_{r(\varphi)}$ at each (reference) location τ_i . Figure 1a illustrates this decomposition.

Direction space $\mathcal{D}_\varphi$. In the space $\mathcal{S} \subset \mathbb{R}^2$, the direction is represented by the angle φ relative to a reference direction. We define the reference direction as the latitude lines, i.e. the 0° direction. Thus, the direction space $\mathcal{D}_\varphi$ is a circular domain $[0, 2\pi)$ for any location. Fixing a location τ_i , any other location $\tau_j \neq \tau_i$ has a unique direction with respect to τ_i .

Distance space $\mathcal{D}_{r(\varphi)}$. The other subspace, i.e. the distance space $\mathcal{D}_{r(\varphi)}$, is determined by the direction φ . Its size is the distance from τ_i to the boundary of $\mathcal{S}$ in the direction φ , which is a function of φ . Since $\mathcal{S}$ is rectangular, $r(\varphi)$ has a closed-form expression, which is detailed later. Thus, fixing τ_i and a direction φ , any other location τ_j has a unique distance $r(\varphi)$ with respect to τ_i .

By this decomposition, we can represent any location $\tau_j \in \mathcal{S}$ as coordinates $(\varphi, r(\varphi)) \in (\mathcal{D}_\varphi, \mathcal{D}_{r(\varphi)})$ relative to τ_i . As an example, τ_j in Figure 1a is represented by $(\varphi_1, r(\varphi_1))$. This representation is unique and reversible: any pair of coordinates $(\varphi, r(\varphi))$ can be mapped back to a unique location $\tau_j \in \mathcal{S}$. Thus, if τ_j is a sensitive

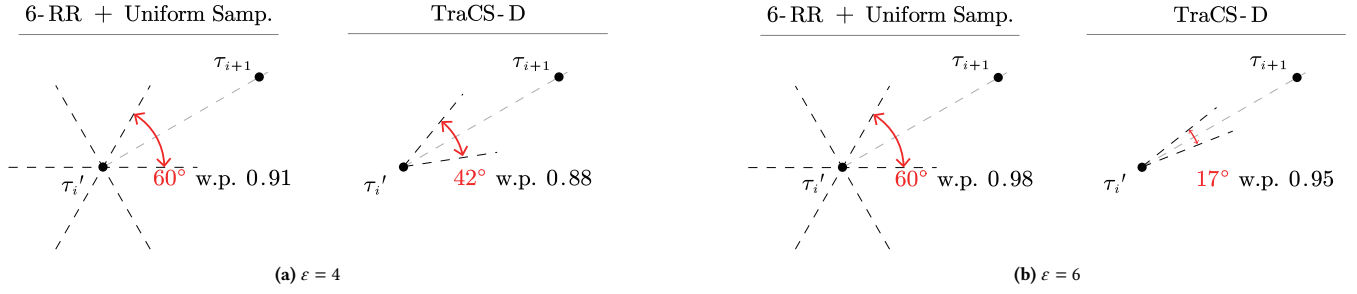


Figure 2: Comparison of the dominant sectors of the strawman approach and TraCS-D with $\epsilon = 4$ and $\epsilon = 6$. The red angular arcs indicate the dominant sectors, with their probabilities shown on the right. The dominant sector of TraCS-D narrows as ϵ increases, leading to a smaller inner-sector (sampling) error. In contrast, the strawman approach has a fixed dominant sector, which is independent of ϵ and leads to a large inner-sector error.

location, an LDP mechanism applied to $\tau_j = (\varphi, r(\varphi))$ is essentially applied to φ and $r(\varphi)$. By Composition Theorem 1, we can design perturbation mechanisms for φ and $r(\varphi)$, respectively, to achieve LDP for τ_j .

3.2.1 Direction Perturbation. The technical challenge in direction perturbation is designing an LDP mechanism for the circular domain $\mathcal{D}_\varphi = [0, 2\pi)$. This is non-trivial due to the following properties of the circular domain:

- The circular domain $[0, 2\pi)$ is bounded but not finite, making the unbounded mechanisms and discrete mechanisms inapplicable.
- The circular domain has a unique distance metric, e.g. the distance between 0 and 2π is 0, not 2π , making the Euclidean distance-based mechanisms inapplicable.

Specifically, classical mechanisms like Laplace and Gaussian [15] add unbounded noise, resulting in an output domain of $(-\infty, +\infty)$. This output domain makes them inapplicable to the circular domain $[0, 2\pi)$. Additionally, these mechanisms are designed for the distance metric $d(y, x) = |y - x|$, which is inconsistent with the distance in circular domain. Discrete mechanisms, such as k -RR [21], assume a finite domain of size k and therefore cannot be directly applied to the continuous circular domain $[0, 2\pi)$. Recently, OGPM [47] proposed a piecewise-based mechanism tailored to circular domains, which inspires our design.

To highlight the necessity and superiority of TraCS-D's piecewise-based design compared to simpler alternatives, we start by presenting a strawman approach that extends the k -RR mechanism to the circular domain $[0, 2\pi)$.

Strawman approach: k -RR + uniform sampling. A former work ATP [44] divides the direction space $[0, 2\pi)$ into k sectors and applies the k -RR mechanism. This approach ensures LDP for the k sectors but not for the entire $[0, 2\pi)$ domain. Even so, we can extend this approach to $[0, 2\pi)$ by further uniformly sampling a direction φ' from the output sector of k -RR.

Specifically, the circular domain $[0, 2\pi)$ is divided into k sectors $[i-1, i) \cdot 2\pi/k$ for $i = 1, 2, \dots, k$. Assume the sensitive direction φ falls into the j -th sector. Applying the k -RR mechanism outputs sector $i = j$ with probability $p = \exp(\epsilon)/(k-1+\exp(\epsilon))$, or outputs sector $i \neq j$ with probability $p/\exp(\epsilon)$. Then, we uniformly sample a direction φ' from the i -th sector, i.e. $\varphi' \sim U(i-1, i) \cdot 2\pi/k$.

Limitation. This strawman approach introduces inherent *inner-sector* errors due to uniform sampling within a specific sector. That is, even though the perturbed direction φ' falls into the same sector as the sensitive direction φ , the distance between φ and φ' may still be large due to the large sector size. This issue is particularly prominent when k is small, as each sector then spans $2\pi/k$ radians. Uniform sampling within such wide sectors introduces substantial inner-sector errors, which persist regardless of how large the privacy parameter ϵ is set. On the other hand, increasing k to reduce sector width impairs the utility of the k -RR mechanism [38].

Design rationale. To address this limitation, we design a direction perturbation mechanism in which the perturbed direction is independent of hyperparameters like k , thereby avoiding inherent errors beyond those introduced by the privacy parameter ϵ .

We adapt the design of piecewise-based mechanisms for circular domains from OGPM [47]. Specifically, (i) we instantiate a piecewise-based mechanism over the circular domain $[0, 2\pi)$, which guarantees LDP for the entire direction space; (ii) the perturbed direction is sampled from a piecewise probability distribution, which is centered around the sensitive direction with high probability and depends solely on the privacy parameter ϵ , thus eliminating the need for pre-defined fixed sectors.

DEFINITION 5 (DIRECTION PERTURBATION MECHANISM). Given a sensitive direction φ and a privacy parameter ϵ , TraCS-D's direction perturbation mechanism $\mathcal{M}_o : [0, 2\pi) \rightarrow [0, 2\pi)$ is defined by:

$$pdf[\mathcal{M}_o(\varphi) = \varphi'] = \begin{cases} p_\epsilon & \text{if } \varphi' \in [l_{\varphi, \epsilon}, r_{\varphi, \epsilon}), \\ p_\epsilon / \exp(\epsilon) & \text{otherwise,} \end{cases}$$

where $p_\epsilon = \frac{1}{2\pi} \exp(\epsilon/2)$ is the sampling probability, and $[l_{\varphi, \epsilon}, r_{\varphi, \epsilon})$ is the sampling interval that

$$l_{\varphi, \epsilon} = \left(\varphi - \pi \frac{\exp(\epsilon/2) - 1}{\exp(\epsilon) - 1} \right) \bmod 2\pi,$$

$$r_{\varphi, \epsilon} = \left(\varphi + \pi \frac{\exp(\epsilon/2) - 1}{\exp(\epsilon) - 1} \right) \bmod 2\pi.$$

The above mechanism $\mathcal{M}_o$ is defined by a piecewise probability distribution with three pieces, where the central piece $[l_{\varphi, \epsilon}, r_{\varphi, \epsilon})$ has a higher probability density p_ϵ . As a piecewise-based mechanism, it evidently satisfies LDP for the circular domain $[0, 2\pi)$; see Appendix B.1 for the proof.

A key advantage of TraCS-D's direction perturbation mechanism is the "dominant" sector. We refer to the central piece $[l_{\varphi,\varepsilon}, r_{\varphi,\varepsilon})$ as the dominant sector because it centers around the sensitive direction φ with high probability. The dominant sector adapts dynamically to the privacy parameter ε : as ε increases, the dominant sector becomes narrower and more tightly centered around the sensitive direction φ . This adaptivity effectively mitigates the inner-sector errors inherent in the strawman approach, where the sector width is fixed and independent of ε .

EXAMPLE 1. Figure 2 compares the strawman approach and TraCS-D. Assume the next location τ_{i+1} is a sensitive location needing direction perturbation, thus the sensitive direction is $\varphi : \tau_i' \rightarrow \tau_{i+1}$. The dominant sector of TraCS-D is much smaller as ε increases. For instance, when $\varepsilon = 6$ and $\varphi = \pi/6$, it can be calculated that $p_\varepsilon \approx 3.20$ and $[l_{\varphi,\varepsilon}, r_{\varphi,\varepsilon}) \approx [0.119\pi, 0.214\pi)$. Thus, the dominant sector has a size of $0.095\pi \approx 17^\circ$ with probability $0.095\pi \times 3.20 \approx 0.95$ of being chosen. In comparison, the strawman approach has a dominant sector of $2\pi/6 = 60^\circ$ with probability 0.98 of being chosen. With almost the same probability of being chosen, TraCS-D's dominant sector is significantly narrower than that of the strawman approach, resulting in a much smaller inner-sector (sampling) error.

Detailed comparison between the strawman approach and Definition 5 is provided in Appendix C.4. This Appendix will show that TraCS-D achieves a better trade-off between the size and probability of the dominant sector compared to the strawman approach with any k value.

3.2.2 Distance Perturbation. The other subspace of TraCS-D is the distance space $\mathcal{D}_{r(\varphi)}$. Besides the boundedness, the size of this space also varies with the reference location τ_i and the direction φ . The technical challenge in designing an LDP mechanism for this space lies in calculating the size $|\mathcal{D}_{r(\varphi)}|$ for any φ at any τ_i .

Given location τ_i , any other location $\tau_j \in \mathcal{S}$ can be represented by a pair of coordinates $(\varphi, r(\varphi))$ relative to τ_i . Since we focus on $\mathcal{S}$ as a rectangular area $[a_{\text{sta}}, a_{\text{end}}] \times [b_{\text{sta}}, b_{\text{end}}]$, the distance from τ_i to the boundary of $\mathcal{S}$ has a closed-form expression depending on the direction φ . Specifically, this distance falls into one of the following four cases:

$$|\mathcal{D}_{r(\varphi)}| \in \left\{ \frac{a_{\text{end}} - a_i}{\cos(\varphi)}, \frac{b_{\text{end}} - b_i}{\sin(\varphi)}, \frac{a_i - a_{\text{sta}}}{-\cos(\varphi)}, \frac{b_i - b_{\text{sta}}}{-\sin(\varphi)} \right\}, \quad (1)$$

depending on the value of φ . Appendix C.5 provides the detailed equation form of Equation (1). Figure 3 illustrates the first two cases of $|\mathcal{D}_{r(\varphi)}|$: when φ falls into $[0, \varphi_1)$, i.e. Figure 3a, $|\mathcal{D}_{r(\varphi)}| = (a_{\text{end}} - a_i)/\cos(\varphi)$; Figure 3b shows the second case, i.e. when $\varphi \in [\varphi_1, \varphi_2)$, then $|\mathcal{D}_{r(\varphi)}| = (b_{\text{end}} - b_i)/\sin(\varphi)$.

With the known distance space $[0, |\mathcal{D}_{r(\varphi)}|]$ and sensitive distance $r(\varphi)$, we can design a distance perturbation mechanism to guarantee LDP for $\mathcal{D}_{r(\varphi)}$. For the convenience of presentation, we normalize the distance $r(\varphi)$ by $|\mathcal{D}_{r(\varphi)}|$, resulting in a normalized distance $\bar{r}(\varphi) \in [0, 1)$.¹¹ We then employ the piecewise-based mechanism in OGPM [47] for the normalized distance $\bar{r}(\varphi)$ to guarantee LDP on $[0, 1)$.

¹¹We use $[0, 1)$ instead of $[0, 1]$ to ease the presentation of Definition 6 and to align with the circular domain. These two domains are equivalent in implementation.

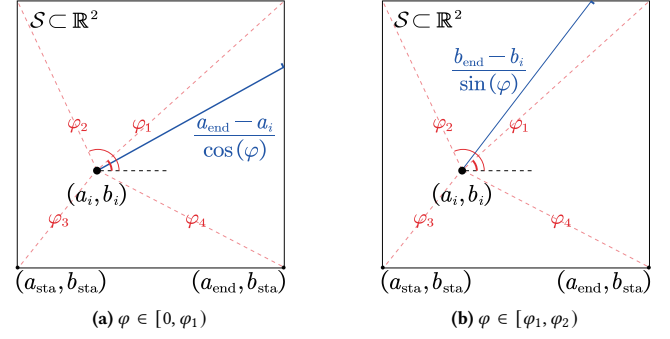


Figure 3: Two cases of $|\mathcal{D}_{r(\varphi)}|$ at τ_i (blue lines).

DEFINITION 6 (DISTANCE PERTURBATION MECHANISM). Given a sensitive distance $\bar{r}(\varphi)$ and a privacy parameter ε , TraCS-D's distance perturbation mechanism $\mathcal{M}_-$: $[0, 1) \rightarrow [0, 1)$ is defined by:

$$\text{pdf}[\mathcal{M}_-(\bar{r}(\varphi)) = \bar{r}'(\varphi)] = \begin{cases} p_\varepsilon & \text{if } \bar{r}'(\varphi) \in [u, v), \\ p_\varepsilon / \exp(\varepsilon) & \text{otherwise,} \end{cases}$$

where $p_\varepsilon = \exp(\varepsilon/2)$ is the sampling probability, and $[u, v)$ is the sampling interval that

$$[u, v) = \begin{cases} \bar{r}(\varphi) + [-C, C) & \text{if } \bar{r}(\varphi) \in [C, 1 - C), \\ [0, 2C) & \text{if } \bar{r}(\varphi) \in [0, C), \\ [1 - 2C, 1) & \text{otherwise,} \end{cases}$$

with $C = (\exp(\varepsilon/2) - 1) / (2 \exp(\varepsilon) - 2)$.

Similar to $\mathcal{M}_\circ$, the above mechanism $\mathcal{M}_-$ has a higher probability density p_ε in the central piece $[u, v)$, and a larger ε results in a higher p_ε and a narrower $[u, v)$, boosting the utility. It ensures LDP for $[0, 1)$ and outputs a perturbed normalized distance $\bar{r}'(\varphi)$. To cooperate with the direction perturbation, $\bar{r}'(\varphi)$ can be mapped to another distance space $\mathcal{D}_{r(\varphi')}$ at the perturbed direction φ' . This is a linear mapping without randomness, so the post-processing property [15] ensures it preserves the same privacy level.

3.2.3 Workflow of TraCS-D. Combining the direction perturbation and distance perturbation, TraCS-D perturbs each sensitive location in the trajectory.

Algorithm 1 presents the workflow of TraCS-D. It takes the location space $\mathcal{S}$, the sensitive trajectory $\mathcal{T}$, and the privacy parameter ε as input. Since TraCS-D relies on a non-sensitive reference location, we add a dummy location τ_0' to $\mathcal{T}$, which can be the starting coordinate of $\mathcal{S}$ or a randomly drawn location. Then, TraCS-D iterates over each pair of consecutive locations τ_i' and τ_{i+1} (line 3). For each sensitive location τ_{i+1} , it perturbs the direction (red block) and normalized distance (blue block) with privacy parameters ε_d and $\varepsilon - \varepsilon_d$, respectively. The perturbed distance is then mapped along the perturbed direction (line 9). Line 11 updates the reference location to the perturbed location τ_{i+1}' , which is then used as the reference for the next iteration ($i + 1$). Finally, the algorithm outputs the perturbed trajectory $\mathcal{T}'$.

Algorithm 1 uses each τ_i' as the reference location for the next perturbation. In fact, it can be any non-sensitive location in $\mathcal{S}$. We choose different τ_i' for alleviating the impact of a specific reference location.

Algorithm 1: TraCS-D

Input: Rectangular location space $\mathcal{S}$, sensitive trajectory $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$, privacy parameter ε

Output: Perturbed trajectory $\mathcal{T}' = \{\tau'_1, \tau'_2, \dots, \tau'_n\}$

```

1  $\mathcal{T}' \leftarrow \emptyset, \mathcal{T} \leftarrow \tau'_0 \cup \mathcal{T};$  ▷ Add a dummy location  $\tau'_0$ 
2 for  $i \leftarrow 0$  to  $n - 1$  do
   ▷  $\tau'_i$  is the ref location for this iteration
3    $\tau'_i = (a_i, b_i), \tau_{i+1} = (a_{i+1}, b_{i+1});$ 
   ▷ Sensitive direction
4    $\varphi \leftarrow \text{atan2}(b_{i+1} - b_i, a_{i+1} - a_i);$ 
   ▷ Perturb direction
5    $\varphi' \leftarrow \mathcal{M}_o(\varphi; \varepsilon_d);$ 
6    $R \leftarrow |\mathcal{D}_r(\varphi)|$  in Equation (1);
7    $\bar{r}(\varphi) \leftarrow \|\tau_{i+1} - \tau'_i\|_2 / R;$  ▷ Sensitive (norm.) distance
8    $\bar{r}'(\varphi) \leftarrow \mathcal{M}_-(\bar{r}(\varphi); \varepsilon - \varepsilon_d);$  ▷ Perturb distance
9    $R' \leftarrow |\mathcal{D}_r(\varphi')|, r'(\varphi') \leftarrow \bar{r}'(\varphi) \times R';$  ▷ De-normalize
   ▷ Transform back to (longitude, latitude)
10   $\tau'_{i+1} = (a_i + r'(\varphi') \cos(\varphi'), b_i + r'(\varphi') \sin(\varphi'));$ 
11   $\mathcal{T}' \leftarrow \mathcal{T}' \cup \tau'_{i+1};$  ▷  $\tau'_{i+1}$  is the next ref location
12 return  $\mathcal{T}';$ 

```

3.2.4 Analysis of TraCS-D. This subsection analyzes the privacy, computational complexity, and utility of TraCS-D.

THEOREM 2. *TraCS-D (Algorithm 1) satisfies $n\varepsilon$ -LDP for the rectangular location space $\mathcal{S}$.*

PROOF. (Sketch) The direction perturbation mechanism $\mathcal{M}_o$ satisfies ε_d -LDP by its definition. For the distance perturbation, the randomness comes entirely from mechanism $\mathcal{M}_-$, which satisfies $(\varepsilon - \varepsilon_d)$ -LDP. The post-processing property preserves the same privacy level after linearly mapping the perturbed distance. Then each location satisfies ε -LDP by Composition Theorem 1 (or by computing the 2D *pdf* ratio in the LDP definition). Hence, the entire perturbed trajectory satisfies $n\varepsilon$ -LDP. Appendix B.2 provides details. $\square$

Complexity. TraCS-D (Algorithm 1) has $\Theta(1)$ time complexity for each location, as each line is $\Theta(1)$. Thus, the entire algorithm has $\Theta(n)$ time complexity, where n is the length of the trajectory. The space complexity is also $\Theta(n)$ because it needs to store the perturbed trajectory $\mathcal{T}'$.

The $\Theta(n)$ time complexity is optimal for per-location perturbation mechanisms. Among existing LDP mechanisms for discrete location spaces, even if we ignore the complexity of the score function and sampling in the Exponential mechanism [31], NGRAM [10] has $\Theta(tn)$ time complexity, where t is the number of predefined “gram”. It also includes a large constant factor due to its search to satisfy reachability constraints. L-SRR [35] has $\Theta(mn)$ time complexity, where m is the number of locations, due to the grouping of locations. ATP [44] also has $\Theta(mn)$ time complexity, because of its trajectory merging step.

Note that the $\Theta(n)$ space complexity is for the convenience of presenting the algorithm. It can be reduced to $\Theta(1)$ by only updating each perturbed location τ'_{i+1} in place within $\mathcal{T}$. This results in a lightweight memory footprint, which is crucial for edge computing.

THEOREM 3. *In TraCS-D, both the worst-case mean square error (MSE) of $\mathcal{M}_o$ and $\mathcal{M}_-$ converge to zero with a rate of $\Theta(e^{-\varepsilon/2})$.*

PROOF. (Sketch) The MSE can be calculated by the closed-form expression of $\mathcal{M}_o$ and $\mathcal{M}_-$, allowing us to prove the convergence rate. Appendix B.3 provides the details. $\square$

This theorem indicates that both the error of the perturbed direction and distance are reduced exponentially with the privacy parameter ε . Note that the size of the distance space also affects the variance. Specifically, denote the output of $\mathcal{M}_-$ as a random variable Y and the sensitive input as x . After the linear mapping, the perturbed distance is $Y \cdot |\mathcal{D}_r|$ and the sensitive distance is $x \cdot |\mathcal{D}_r|$, where $|\mathcal{D}_r|$ is the size of the distance space. Thus, the MSE of the perturbed distance is $\text{MSE}[Y \cdot |\mathcal{D}_r|] = |\mathcal{D}_r|^2 \cdot \text{MSE}[Y]$. Therefore, a smaller $|\mathcal{D}_r|$ leads to a smaller MSE of the perturbed distance.

3.3 TraCS-C

TraCS-C decomposes the rectangular location space $\mathcal{S}$ into two independent distance spaces $\mathcal{D}_a$ and $\mathcal{D}_b$, as shown in Figure 1b. Specifically, we can fix $(a_{\text{sta}}, b_{\text{sta}})$ as the reference location, with longitude a_{sta} as the a -axis and latitude b_{sta} as the b -axis. Then any location $\tau_i \in \mathcal{S}$ can be represented as a pair of Cartesian coordinates (d_a, d_b) , where $d_a \in \mathcal{D}_a$ and $d_b \in \mathcal{D}_b$ are the distances from τ_i to the a -axis and b -axis, respectively. This representation is unique and reversible: any pair of distances (d_a, d_b) can be mapped back to a unique location $\tau_i \in \mathcal{S}$. Thus, if τ_i is a sensitive location, an LDP mechanism applied to τ_i is essentially applied to d_a and d_b . By Composition Theorem 1, we can achieve LDP for τ_j by applying perturbation mechanisms for d_a and d_b , respectively.

3.3.1 Coordinates Perturbation. In TraCS-C’s representation of locations, $\mathcal{D}_a = [0, a_{\text{end}} - a_{\text{sta}})$ and $\mathcal{D}_b = [0, b_{\text{end}} - b_{\text{sta}})$ are independent of any specific location. Therefore, the location perturbation of any τ_i can be performed independently in $\mathcal{D}_a$ and $\mathcal{D}_b$.

Following the same idea of distance perturbation in TraCS-D, we normalize the distance d_a and d_b to $[0, 1)$ by dividing $|\mathcal{D}_a|$ and $|\mathcal{D}_b|$, respectively. Then the mechanism $\mathcal{M}_-$ in Definition 6 provides ε -LDP for the normalized distances, and the linear mappings back to $\mathcal{D}_a$ and $\mathcal{D}_b$ preserve the privacy guarantee.

Algorithm 2 shows the workflow of TraCS-C. It takes the rectangular location space $\mathcal{S}$, the sensitive trajectory $\mathcal{T}$, and the privacy parameter ε as input, and outputs the perturbed trajectory $\mathcal{T}'$. For each location $\tau_i = (a_i, b_i)$ in $\mathcal{T}$, Line 4 computes its coordinates (d_a, d_b) and normalizes them to $(\bar{d}_a, \bar{d}_b) \in [0, 1) \times [0, 1)$. Lines 5–6 perturb the normalized coordinates $\bar{d}_a$ and $\bar{d}_b$ using $\mathcal{M}_-$ with privacy parameter $\varepsilon/2$. Then Line 8 maps the perturbed normalized coordinates back to $\mathcal{D}_a$ and $\mathcal{D}_b$, and converts them to the longitude-latitude representation τ'_i .

3.3.2 Analysis of TraCS-C. Privacy and complexity of TraCS-C can be analyzed similarly to TraCS-D.

THEOREM 4. *TraCS-C (Algorithm 2) satisfies $n\varepsilon$ -LDP for the rectangular location space $\mathcal{S}$.*

PROOF. (Sketch) Algorithm 2 uses mechanism $\mathcal{M}_-$ twice, each with a privacy parameter $\varepsilon/2$. By Composition Theorem 1 (or by

Algorithm 2: TraCS-C

Input: Rectangular location space $\mathcal{S}$, sensitive trajectory $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$, privacy parameter ϵ

Output: Perturbed trajectory $\mathcal{T}' = \{\tau'_1, \tau'_2, \dots, \tau'_n\}$

```

1  $\mathcal{T}' \leftarrow \emptyset;$ 
2 for  $i \leftarrow 1$  to  $n$  do
3    $\tau_i = (a_i, b_i);$ 
4    $\triangleright$  Normalize coordinates
5    $(\vec{d}_a, \vec{d}_b) \leftarrow (\frac{a_i - a_{\text{sta}}}{a_{\text{end}} - a_{\text{sta}}}, \frac{b_i - b_{\text{sta}}}{b_{\text{end}} - b_{\text{sta}}});$ 
6    $\vec{d}'_a \leftarrow \mathcal{M}_-(\vec{d}_a; \epsilon/2);$   $\triangleright$  Perturb coordinates
7    $\vec{d}'_b \leftarrow \mathcal{M}_-(\vec{d}_b; \epsilon/2);$ 
8    $\triangleright$  De-normalize to  $(\mathcal{D}_a, \mathcal{D}_b)$ 
9    $(d'_a, d'_b) \leftarrow (\vec{d}'_a | \mathcal{D}_a, \vec{d}'_b | \mathcal{D}_b);$   $\tau'_i = (a_{\text{sta}}, b_{\text{sta}}) + (d'_a, d'_b);$ 
10   $\mathcal{T}' \leftarrow \mathcal{T}' \cup \{\tau'_i\};$ 
11 return  $\mathcal{T}';$ 

```

computing the 2D *pdf* ratio in the LDP definition), their composition satisfies ϵ -LDP. The subsequent linear mappings to $\mathcal{D}_a$ and $\mathcal{D}_b$ are post-processing steps that preserve the same privacy. Hence, each perturbed location satisfies ϵ -LDP, and the whole trajectory satisfies $n\epsilon$ -LDP. Appendix B.4 provides details. $\square$

Complexity. TraCS-C (Algorithm 2) has $\Theta(n)$ time complexity and $\Theta(n)$ space complexity, where n is the length of the trajectory. The time complexity is $\Theta(n)$ because each line of Algorithm 2 is $\Theta(1)$. The space complexity is $\Theta(n)$ because it needs to store the perturbed trajectory $\mathcal{T}'$. Note that the $\Theta(n)$ space complexity is for the convenience of presenting the algorithm. It can also be reduced to $\Theta(1)$ by updating each perturbed location τ'_i in place within $\mathcal{T}$, which benefits edge devices with limited computation resources.

3.4 Rounding to Discrete Space

Although TraCS is designed for continuous location spaces, it can be applied to any fine-grained discrete location space by rounding the perturbed locations to the nearest discrete locations. The rounding is a post-processing step that does not affect the privacy guarantee for the continuous space.

Specifically, if the discrete location space is a set of cells, we can round the perturbed location to the cell that contains it. Formally, assume $\mathcal{C} = \{c_1, \dots, c_m\} \subseteq \mathcal{S}$ discretize $\mathcal{S}$ into m cells. If τ'_i is the perturbed location of τ_i in TraCS, then the discretized TraCS outputs c_j such that $\tau'_i \in c_j$. Commonly, the cells are evenly divided, so the rounding costs $\Theta(1)$ time, as c_j can be determined by the coordinates of τ'_i .

If the location space is a set of location points, we can round the perturbed location to the nearest location point. Formally, assume $\mathcal{P} = \{p_1, \dots, p_m\} \subseteq \mathcal{S}$ is a set of location points, and τ'_i is the perturbed location of τ_i in TraCS. Then, the discretized TraCS outputs $\arg \min_{p_j \in \mathcal{P}} \|\tau'_i - p_j\|_2$. The time complexity of rounding to the nearest location point is $O(m)$ in the worst case. In practice, this time complexity can be significantly reduced by using heuristic or approximation algorithms.

The candidate locations in a discrete space are typically public knowledge, which enables a heuristic selection of a better reference location for TraCS-D. The key observation is that a smaller distance

space $\mathcal{D}_{r(\varphi)}$ yields a smaller distance-perturbation error. Therefore, it is preferable to choose a reference location that minimizes the size of $\mathcal{D}_{r(\varphi)}$ with respect to the candidate locations.

Specifically, the direction from the reference location to other locations is often concentrated in a dominant sector, as illustrated in Figure 2. If such a sector covers most candidate locations (thus reducing rounding error) and is close to the boundary of the location space (thus shrinking the distance domain), then the corresponding reference location is a good choice.

3.5 Discussions and Extensions

This subsection discusses the comparison between TraCS-D and TraCS-C, Euclidean versus spherical geometry, technical considerations in TraCS, and other discussions and extensions.

3.5.1 Comparison of Privacy Between TraCS-D and TraCS-C. Although TraCS-D and TraCS-C provide the same level of privacy quantified by ϵ -LDP—more specifically, location-level (or event-level) LDP as described in surveys [3, 30]—they have different privacy interpretations.

In TraCS-D, the LDP guarantee is provided for the direction information and the subsequent distance information. This means that when the use of TraCS-D and the perturbed trajectory $\mathcal{T}'$ is publicly known, any observer can hardly infer the sensitive direction $\varphi : \tau'_i \rightarrow \tau_{i+1}$ and the sensitive distance $r(\varphi) : |\tau'_i \rightarrow \tau_{i+1}|$ from the known τ'_i and τ'_{i+1} . Although the distance space $\mathcal{D}_{r(\varphi)}$ relies on a specific direction, it does not leak the direction information, as the direction space $\mathcal{D}_\varphi = [0, 2\pi)$ is independent of any location and TraCS-D uses the perturbed direction.

In TraCS-C, the LDP guarantee is provided for the two independent distance spaces $\mathcal{D}_a$ and $\mathcal{D}_b$. This means that when the use of TraCS-C and the perturbed trajectory $\mathcal{T}'$ is publicly known, any observer can hardly infer the sensitive distances d_a and d_b .

3.5.2 Euclidean Geometry vs Spherical Geometry. TraCS is designed for a rectangular location space $\mathcal{S} \subset \mathbb{R}^2$ under Euclidean geometry, where distances are measured by the Euclidean metric. In particular, TraCS-D requires computing the distance spaces $\mathcal{D}_{r(\varphi)}$, which are defined with respect to the Euclidean distance. This choice is made for generality: (i) for many continuous trajectories (e.g. from wearable sensors or indoor devices), Euclidean geometry is a natural choice; and (ii) for city-scale GPS trajectories (e.g. in Chicago and Tokyo), the distortion induced by approximating geographic coordinates as Cartesian coordinates is typically negligible.

Under spherical geometry, distances are measured by the great-circle distance. (i) For country-scale GPS trajectories, standard map projections such as UTM [34] can convert GPS coordinates into local Cartesian coordinates, after which Euclidean distance can be used. (ii) For general spherical domains (e.g. the Earth's surface), we can treat longitude and latitude as two independent coordinates and redesign TraCS-C via independent composition of $(\mathcal{M}_\circ, \mathcal{M}_\circ)$. Specifically, for a location with GPS coordinates $\tau_i = (a_i, b_i) \in ([-\pi, \pi), [-\pi/2, \pi/2])$, i.e. longitude and latitude, we perturb longitude and latitude independently using $\mathcal{M}_\circ$ and $\mathcal{M}_\circ$, respectively. This design does not involve Euclidean distance and aligns with the semantics of longitude and latitude as separate coordinates.

3.5.3 Why Perturb Direction First in TraCS-D. In TraCS-D, we perturb the direction first and then the distance. This order is motivated by two considerations. First, the direction space $\mathcal{D}_\varphi = [0, 2\pi)$ is location-independent, making it well suited for direct perturbation. Second, the distance space $\mathcal{D}_{r(\varphi)}$ is always defined relative to a specific direction φ . If we were to perturb the distance before the direction, then after perturbing the direction we would need to redefine the corresponding distance domain using the perturbed direction, which complicates the procedure. Perturbing the direction first avoids this issue and yields a cleaner mechanism design.

3.5.4 Impact of the Size of $\mathcal{S}$. Intuitively, a larger location space $\mathcal{S}$ requires larger privacy parameters to achieve the same level of utility. For TraCS-D and TraCS-C, this impact is reflected in the size of the direction space and distance spaces.

In TraCS-D, the direction space $\mathcal{D}_\varphi = [0, 2\pi)$ is independent of the size of $\mathcal{S}$, meaning that the MSE of the perturbed direction, $\text{MSE}[\varphi']$, remains constant regardless of the size of $\mathcal{S}$. The distance space $\mathcal{D}_{r(\varphi)}$ depends on the specific direction φ and the size of $\mathcal{S}$. In the worst case, $\mathcal{D}_{r(\varphi)}$ is the diagonal of $\mathcal{S}$. We have shown that the MSE of $\mathcal{M}_-$ is quadratic to the size of the distance space after linear mapping in Section 3.2.4. Therefore, when $|\mathcal{D}_{r(\varphi)}|$ increases linearly, the expected error of the perturbed distance, i.e. $\mathbb{E}[|r'(\varphi) - r(\varphi)|]$, also increases linearly. Nonetheless, the error in the direction space $\mathcal{D}_\varphi$ affects the error in $\mathcal{D}_{r(\varphi)}$ in a non-linear way. Specifically, under the Euclidean distance metric, this effect is $|\mathcal{D}_{r(\varphi)}| \cdot 2 \sin(|\varphi - \varphi'|/2)$, which is the chord length of the arc between φ and φ' .

In TraCS-C, both distance spaces $\mathcal{D}_a$ and $\mathcal{D}_b$ are determined by the size of $\mathcal{S}$. Since they use mechanism $\mathcal{M}_-$ independently, their expected error increases linearly with the increase of $|\mathcal{D}_a|$ and $|\mathcal{D}_b|$. Meanwhile, the error in $\mathcal{D}_a$ affects the error in $\mathcal{D}_b$ linearly.

3.5.5 Comparison with Geo-indistinguishability. Geo-indistinguishability [2] (Geo-Ind for brevity, also called metric privacy [32] in recent works) is a more general (or weaker) version of LDP. From the perspective of Geo-ind, LDP force the proximity-aware function $d(x, y)$ in Geo-ind to be $d(x, y) = 1$ for all $x, y \in \mathcal{S}$, ignoring that different x and y have different $d(x, y)$ in the context of their locations. Conversely, if the upper bound $\max_{x, y \in \mathcal{S}} d(x, y) \leq d^*$ holds in $\mathcal{S}$, then TraCS also satisfies $d^* \epsilon$ -Geo-ind. However, this trivial satisfaction of $d^* \epsilon$ -Geo-ind is highly suboptimal, since it uses the global upper bound of $d(x, y)$ rather than the fine-grained, actual values of $d(x, y)$ for different location pairs of x and y .

We omit trajectory utility evaluations against Geo-ind as it is a different privacy model from LDP. Generally, a well-designed Geo-ind mechanism can achieve better data utility than LDP mechanisms. For piecewise-shaped mechanisms, it is possible to tailor different pieces to account for varying $d(x, y)$ for different x, y , thereby satisfying Geo-ind while achieving better data utility than the current LDP mechanisms.

3.5.6 Other LDP Mechanisms for Bounded and Continuous Space. In this paper, we employ utility-optimized piecewise-based mechanisms (OGPM) [47] for bounded numerical domains to design the direction and distance perturbations in TraCS. Other LDP mechanisms for bounded numerical domains can also be incorporated into TraCS, but most of them may require redesign to suit the direction perturbation, as the circular domain $\mathcal{D}_\varphi = [0, 2\pi)$ has a

different distance metric than linear domains. One exception is the Purkayastha mechanism [39], which is designed for sphere $\mathbb{S}^{n-1}$, e.g. circular lines when $n = 2$. It can be an alternative to $\mathcal{M}_-$, but not be directly applicable to $\mathcal{S} \subset \mathbb{R}^2$. We don't leverage it for direction perturbation because of its sampling complexity (Section 3.4.2 in [39]) and higher MSE [47].

Another category of LDP mechanisms for bounded numerical domains is truncated mechanisms, including truncated Laplace [17, 26] and truncated Gaussian mechanisms [5]. These mechanisms modify Laplace and Gaussian distributions by redesigning distributions on a bounded domain, e.g. $[0, 1)$. However, truncated mechanisms cannot preserve the same privacy level as their original counterparts [5, 17, 26]. Determining their new privacy level is non-trivial and typically relies on numerical testing algorithms [5, 17] rather than closed-form solutions. Compared with truncated mechanisms, piecewise-based mechanisms are explicitly designed for LDP in bounded numerical domains, making them more effective and easier to analyze.

3.5.7 Other Piecewise-based Mechanisms. In Definition 5 and Definition 6, we use specific instantiations for the parameters p and $[l, r)$ in the piecewise-based mechanism. Besides these instantiations, TraCS can be integrated with any other piecewise-based mechanisms [25, 27, 28, 37, 46] by redesigning them. For example, SW [25, 47] is a famous mechanism proposed for distribution estimation; it is defined on $[0, 1) \rightarrow [-b, 1 + b)$, with b as a parameter. It uses different p and $[l, r)$, and employs a strategy of maximizing the mutual information between the perturbed and sensitive data to determine b . SW can be redesigned [47] for the direction perturbation and distance perturbation in TraCS-D and TraCS-C. Appendix C.6 provides the detailed redesign of SW for them and shows intuitive comparisons with Definition 5 and Definition 6, and Appendix C.7 provides a comparison.

3.5.8 Other Shapes of Location Space. The design of LDP mechanisms is space-specific, implying that extensions to other shapes of location space are not straightforward.

By post-processing. A practical workaround for non-rectangular continuous domains is to post-process perturbed locations so they lie inside the target region. If a perturbed location falls outside the shape, project it to the nearest point inside the shape (for polygons this is the closest point on the boundary). Because the shape is public and projection is a data-independent post-processing step, it does not weaken the LDP guarantee. This approach is especially useful for irregular shapes or unions of multiple cities.

Direct extension. Directly extending TraCS to other shapes requires decomposing the domain into independent subspaces that are compatible with our perturbation primitives. For TraCS, the challenge lies in determining the size of each subspace, i.e. $|\mathcal{D}_\varphi|$, $|\mathcal{D}_{r(\varphi)}|$, $|\mathcal{D}_a|$, and $|\mathcal{D}_b|$. Among them, $|\mathcal{D}_{r(\varphi)}|$ is not apparent but can be calculated for rectangular location spaces. Although calculating $|\mathcal{D}_{r(\varphi)}|$ is generally non-trivial for irregular shapes, TraCS can be extended to parallelogram location spaces by a linear transformation to a rectangular space. Meanwhile, TraCS-D can be extended to circular regions, which are widely used in relative-location representations (e.g. radar and sonar systems). Extending TraCS-C to circular regions is more challenging, since such regions do not

admit a natural Cartesian decomposition into independent coordinates. Appendix C.7 reports experimental results of TraCS-D on circular location spaces.

4 Evaluation

This section evaluates the performance of TraCS on both continuous and discrete spaces.

4.1 Results on Continuous Space

The strawman method—our extension of k -RR for direction perturbation—can be combined with the distance perturbation mechanism in TraCS-D to ensure LDP in continuous spaces $\mathcal{S} \subset \mathbb{R}^2$. For simplicity, we continue to refer to this extended approach as the strawman method, and compare it with TraCS in continuous space. We also include the planar Laplace mechanism [2] + clipping (truncation) as a baseline. This baseline perturbs each location by adding noise drawn from a 2D planar Laplace distribution, and then truncates the output to the location space $\mathcal{S}$ if it falls outside.**

4.1.1 Setup. TraCS requires the following parameters: the location space $\mathcal{S}$, the sensitive trajectory $\mathcal{T}$, the privacy parameter ϵ , and ϵ_d for direction perturbation in TraCS-D. The strawman method also requires an instance of k in k -RR. We use configurations as follows for the experiments.

- **Location space $\mathcal{S}$:** We consider synthetic and real-world settings. (i) The default location space is $[0, 1) \times [0, 1)$. Without loss of generality, we consider a larger space $\mathcal{S} = [0, 2) \times [0, 10)$ for comparison. We generate 100 random trajectories for each location space as synthetic datasets. Specifically, each trajectory is generated by randomly sampling a sequence of 100 points in $\mathcal{S}$. This setup excludes dataset-specific effects to emphasize algorithmic performance. (ii) Areas of TKY and CHI: These two trajectory datasets were collected in Tokyo and Chicago, extracted from the Foursquare dataset [40] and Gowalla dataset [8], respectively. Locations in TKY are within area $[139.47, 139.90) \times [35.51, 35.86)$ and CHI within area $[-87.9, -87.5) \times [41.6, 42.0)$. These two areas are visually shown in Figure 18. We use the first 100 trajectories in each dataset.
- **Privacy parameter ϵ_d :** We set $\epsilon_d = \epsilon\pi/(\pi + 1)$ to heuristically balance the perturbation in $\mathcal{M}_o$ and $\mathcal{M}_d$ according to their domain sizes.**†
- **k -RR:** We set $k = 6$ as the strawman method’s default value, and consider 3-RR and 12-RR for comparison.

We evaluate ϵ in the range $[2, 10]$. For 2D discrete spaces with large location sets, this range represents a moderate privacy level.†† For example, with k -RR mechanism and $|\mathcal{X}| = 3600$ locations, $\epsilon = 10$ yields the true location output with probability ≈ 0.85 , while $\epsilon = 2$ results in a much lower probability of ≈ 0.002 . The Exponential mechanism exhibits similar behavior, though the exact probabilities depend on the score function. In such cases where $|\mathcal{X}|$ is large, even though the probability ratio $\exp(\epsilon)$ in the LDP definition is high,

**Appendix C.8 provides more details on the planar Laplace mechanism + clipping.

††Appendix C.9 provides more discussion on setting ϵ_d .

††In Definition 1, the privacy parameter ϵ is defined with respect to a specific domain $\mathcal{X}$; its practical strength therefore depends on the size of $\mathcal{X}$.

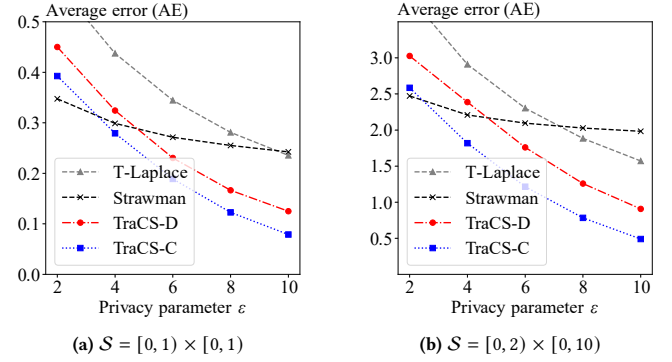


Figure 4: Comparison on synthetic datasets.

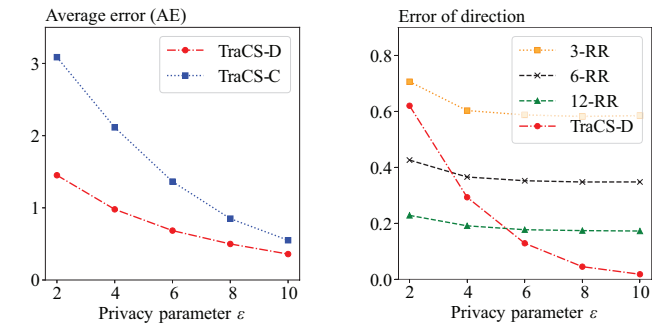


Figure 5: Condition TraCS-D outperforms TraCS-C.

Figure 6: TraCS-D vs k -RR + uniform sampling.

it remains difficult for an adversary to infer the true location with high probability due to the vast number of possible locations.

Trajectory utility metrics. We follow the Euclidean distance to measure the utility of the perturbed trajectory [10, 35, 44]. The distance between two locations τ_i and τ'_i is defined as $\|\tau_i - \tau'_i\|_2$. Therefore, given a sensitive trajectory $\mathcal{T}$ and a perturbed trajectory $\mathcal{T}'$, the average error (AE) among all locations in the trajectory is:

$$AE(\mathcal{T}, \mathcal{T}') = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \|\tau_i - \tau'_i\|_2,$$

where $|\mathcal{T}|$ is the number of locations in the trajectory. A smaller AE indicates better trajectory utility. We compute the average AE across all trajectories in the dataset for comparison. Although AE is not a perfect utility measure, e.g. it may not fully reflect the requirements of specific downstream tasks, it remains the commonly used metric for evaluating the utility of perturbed trajectories [10, 30, 35, 44, 45]. Moreover, many other trajectory utility metrics are directly or indirectly related to AE, such as the preservation of range queries and hotspots [10, 44]. For brevity, we defer these additional metrics to Section 4.2.5.

4.1.2 Error Comparison on Synthetic Datasets. Figure 4 presents the error comparison between TraCS and the strawman method. We can observe the error advantage of TraCS over the truncated Laplace mechanism (T-Laplace) across all ϵ values, and over the strawman method as the privacy parameter ϵ increases. Specifically,

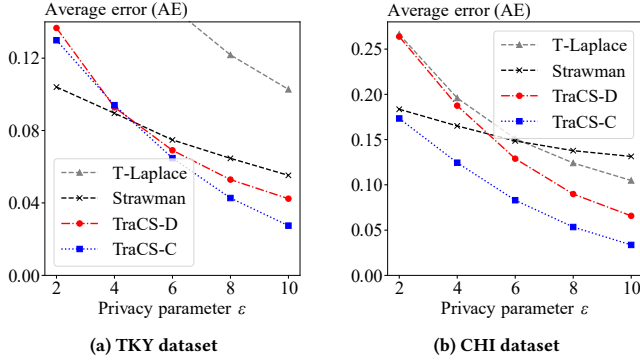


Figure 7: Comparison on real-world datasets.

in Figure 4a, TraCS-C exhibits smaller AE when $\epsilon \geq 3$, and TraCS-D exhibits smaller AE when $\epsilon \geq 5$. When the location space expands to $[0, 2) \times [0, 10)$, i.e. Figure 4b, the advantage of TraCS becomes more significant. Among TraCS, TraCS-C consistently has a smaller AE than TraCS-D. This is because TraCS-D often deals with larger distance spaces $\mathcal{D}_{r(\varphi)}$, which also magnifies the error of direction perturbation. Statistically, the mean AE of TraCS-D is 91.1% of the strawman method in Figure 4a and 86.6% in Figure 4b across all the ϵ values. For TraCS-C the corresponding ratios are 75.5% in Figure 4a and 64.0% in Figure 4b.

Conditions TraCS-D outperforms TraCS-C. The error of TraCS-C depends on the size of the distance spaces $\mathcal{D}_a$ and $\mathcal{D}_b$, while the error of TraCS-D depends only on the size of the distance space $\mathcal{D}_{r(\varphi)}$. Therefore, if $\mathcal{D}_{r(\varphi)}$ is majorly smaller than $\mathcal{D}_a$ or $\mathcal{D}_b$, TraCS-D will outperform TraCS-C. We conduct experiments to verify this condition. Specifically, we set $\mathcal{S} = [0, 2) \times [0, 10)$, and choose a sensitive trajectory along the x-axis, i.e. the short side of $\mathcal{S}$. Then we perturb the trajectory 1000 times using TraCS-C and TraCS-D and calculate the average AE. In this case, $|\mathcal{D}_{r(\varphi)}| \approx 2$, much smaller than $|\mathcal{D}_b| = 10$. Consequently, TraCS-D exhibits smaller AE than TraCS-C across all the ϵ values, as shown in Figure 5. In this experiment, the mean AE of TraCS-D is 49.8% of TraCS-C across all the ϵ values.

Error of direction perturbation. Figure 6 shows the average error of direction perturbation in TraCS-D compared with the strawman method. We collect the sensitive directions of the trajectories in the location space $\mathcal{S} = [0, 1) \times [0, 1)$, and perform direction perturbation using TraCS-D and the strawman method. For the strawman method, we use k -RR with $k = 3, 6, 12$. This experiment reflects empirical error results of mechanism $\mathcal{M}_0$ in Definition 5 and the k -RR + uniform sampling. We can observe an exponentially decreasing error of TraCS-D as ϵ increases, as stated in Theorem 3. In contrast, the strawman method exhibits an eventually stable error as ϵ increases, due to the inherent inner-sector error of k -RR.

4.1.3 Error Comparison on Real-world Datasets. Figure 7 shows the error comparison on the TKY and CHI areas. The trends largely mirror those observed on the synthetic datasets. T-Laplace incurs substantially larger AE than TraCS and the strawman method on the TKY dataset. Across all ϵ values, the mean AE of TraCS-D is

96.8% of the strawman method on TKY and 94.5% on CHI, for TraCS-C the corresponding ratios are 89.7% and 61.2%. The performance gap is smaller on TKY than on CHI, because TKY's location space is more compact, which lowers the error values for all perturbation methods.

4.1.4 Choose TraCS-D or TraCS-C. From the above experiments, we can conclude criteria for choosing TraCS-D or TraCS-C: (i) TraCS-C generally has better trajectory utility than TraCS-D for random trajectories; (ii) TraCS-D is better for specific trajectories where their distance spaces are smaller than in TraCS-C, and suitable for circular areas.

4.2 Results on Discrete Space

Our approach can be applied to discrete spaces by rounding each perturbed location to its nearest discrete point. Importantly, the privacy guarantee is established in the underlying continuous domain and therefore still holds after rounding, regardless of the chosen discretization. In the following, we evaluate the performance of TraCS on discrete spaces and compare it with existing methods.

4.2.1 Setup. Our evaluation includes the following LDP methods for collecting trajectories in discrete spaces:

- NGram [10]: It is often impractical to acquire reachability knowledge in practice [44]. We instead consider a strong reachability constraint, i.e. from each location, the next location can only be within a distance of $0.75 \cdot \hat{a}$,^{§§} where $\hat{a}$ is the maximal length of the location space. This constraint significantly reduces the location space for perturbation, especially for the locations near the boundary.
- L-SRR [35]: We use group number $g = 3$, the same as the empirical optimal value in their paper and code. Locations in the first group are within $0.3 \cdot \hat{a}$ distance from the sensitive location, where $\hat{a}$ is the maximal length of the location space. The second group is within $0.6 \cdot \hat{a}$ distance from the sensitive location, and the remaining locations are in the third group.
- ATP [44]: We set the k -RR parameter $k = 6$ in ATP. The privacy parameter ϵ in original ATP is for the entire trajectory, we distribute it to each location's perturbation according to their algorithms.
- TraCS: We use the same privacy parameter ϵ_d as in the continuous space experiments. For each perturbed location from TraCS, we round it to the nearest discrete location.

For a fair comparison, we use ϵ as the privacy parameter for each location in all methods. The comparison is conducted on the following datasets:

- Synthetic datasets: We uniformly discretize $[0, 1) \times [0, 1)$ into m discrete points and treat the m points as the discrete location space. Specifically, we set $m = 10 \times 10$ and $m = 60 \times 60$ for comparison. Along with them, we generate 100 random trajectories with 100 locations each as synthetic datasets, i.e. each location is randomly sampled from the m discrete points and connected sequentially.

^{§§}While the original NGram employs much stricter time-reachability constraints (e.g. the next location must be within $0.1h \cdot 4\text{km/h}$), this constraint is also known to the adversary, which undermines the privacy guarantee. TraCS are LDP mechanisms guaranteeing privacy for the entire domain and do not incorporate such constraints.

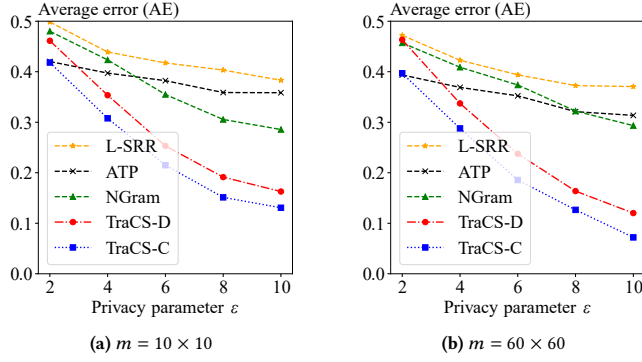


Figure 8: Comparison on synthetic datasets. m is the number of discrete points in the location space.

- TKY and CHI (visualized in Figure 18): We treat every location in the TKY and CHI datasets as a discrete point, which together constitute the discrete location space. TKY contains 7,798 discrete locations; CHI contains 1,000 discrete locations. We use the first 100 trajectories for evaluation.

For the TKY and CHI datasets, we perform the perturbation 5 times for each trajectory and compute the average AE, which is the same setup as in ATP [44].

4.2.2 Error Comparison on Synthetic Datasets. Figure 8 presents the error comparison on the synthetic dataset. Figure 8a shows results for $m = 10 \times 10$, and Figure 8b shows results for $m = 60 \times 60$. We observe that TraCS-C consistently outperforms NGram, L-SRR, and ATP across all the ϵ values, and TraCS-D also outperforms them when $\epsilon \geq 3$. Moreover, the performance advantage of TraCS becomes more significant as m increases to 60×60 . This is because finer-grained discretization allows TraCS’s rounding to align with the sensitive locations more accurately. Statistically, the mean AE of TraCS-C is 66.1% of NGram, 57.1% of L-SRR, and 63.8% of ATP in Figure 8a, and the proportions are 57.6%, 52.6%, and 61.1% in Figure 8b. The mean AE of TraCS-D is 76.9% of NGram, 66.4% of L-SRR, and 74.2% of ATP in Figure 8a, and the proportions are 71.2%, 65.0%, and 75.5% in Figure 8b. These results indicate the advantage of TraCS for synthetic trajectories in discrete spaces.

4.2.3 Error Comparison on Real-world Datasets. Figure 9 presents the error comparison on the TKY and CHI datasets. Compared to the synthetic dataset, NGram, L-SRR, and ATP achieve lower errors, particularly when ϵ is small. However, as ϵ increases, TraCS still outperforms them.

Reasons for the results. (i) ATP performs exceptionally well when ϵ is small, particularly on the TKY dataset. This is because this dataset contains many “condensed” locations, i.e. no other locations near the trajectory. In such cases, if the bi-direction perturbation of ATP is accurate, the perturbed location is error-free. (ii) The AE of ATP does not decrease significantly as ϵ increases, which is consistent with their paper. NGram shows a similar trend, especially on the TKY dataset. This is due to the low performance of the Exponential mechanism for large location spaces, particularly when the locations are concentrated in a small area. In such cases, the score function struggles to differentiate the scores among the

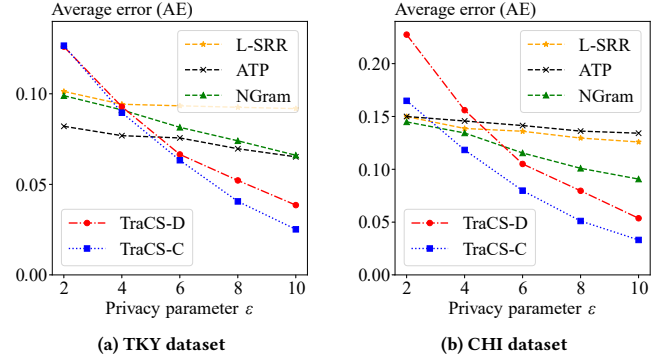


Figure 9: Comparison on real-world datasets. TraCS operates on rectangular areas encompassing the city, which may result in larger AE when ϵ is small.

Table 3: Time cost comparison (in milliseconds).

	ATP	NGram	L-SRR	TraCS-D	TraCS-C
Total	145.7	100.9	6.2	0.06	0.05
Perturb	125.8	92.8	0.003	0.018	0.003

locations. L-SRR exhibits a similar trend. (iii) TraCS exhibits larger AE than them when ϵ is small because it focuses on the rectangular area formed by the longitude and latitude, instead of a set of discrete locations in the cities (visualized in Figure 18). Therefore, the perturbed locations may be far from the city center (where benefits discrete mechanisms) when ϵ is small.

4.2.4 Time Cost Comparison. Table 3 presents the time cost comparison on the TKY dataset. The time cost is measured in milliseconds and averaged per location. It shows that TraCS-D and TraCS-C are significantly faster than NGram, ATP, and L-SRR. Compared to NGram and ATP, this efficiency is primarily due to the piecewise-based mechanisms in TraCS, which require negligible time for perturbation compared to the Exponential mechanism. Compared to L-SRR, TraCS-D and TraCS-C are also faster because they do not require grouping locations.

ATP has the highest time cost because it needs to process two copied trajectories and merges them. NGram has a lower time cost than ATP but is still significantly higher than TraCS. The perturbation procedure accounts for 86.3% and 92.0% of the total time cost in ATP and NGram, respectively. L-SRR has the lowest time cost among the three state-of-the-art methods due to its efficient sampling mechanism, however, its grouping procedure still incurs a non-negligible time cost.

The total time cost of TraCS is less than 0.05% of ATP and NGram, and less than 1% of L-SRR. The perturbation procedure of TraCS-D and TraCS-C only takes 30% and 6% time, respectively. TraCS-D has a higher time cost than TraCS-C because it needs to calculate the distance space $\mathcal{D}_{r(\phi)}$ for each location in the trajectory, while TraCS-C can use the same $\mathcal{D}_a$ and $\mathcal{D}_b$ for all locations.

4.2.5 Experimental Results for Other Metrics. This subsection evaluates the performance of TraCS and existing methods under two

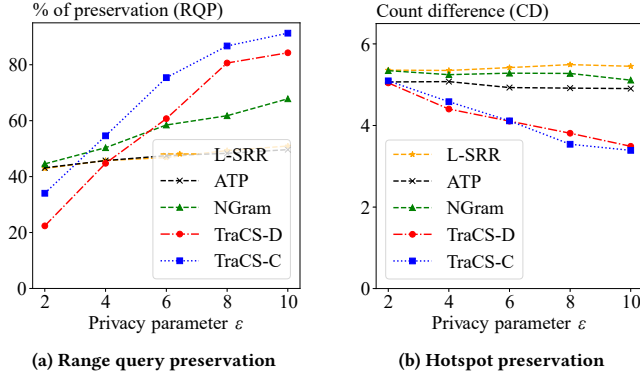


Figure 10: Comparison of other trajectory metrics on the CHI dataset. Higher RQP and lower ACD indicate better performance.

additional metrics: range query preservation (Formula 17 in [10]) and hotspot preservation (Section 6.2.4 in [44]). Given a threshold δ , a perturbed location is considered correct if it lies within a δ distance of the corresponding sensitive location. Formally, for a trajectory $\mathcal{T}$ and its perturbed trajectory $\mathcal{T}'$, the range query preservation (RQP) is defined as:

$$RQP(\mathcal{T}, \mathcal{T}') = \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} \mathbf{1}\{\|\tau_i - \tau'_i\|_2 \leq \delta\} \cdot 100\%,$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. A higher RQP indicates better trajectory utility. Hotspot preservation measures how many hotspots from a given set remain after perturbation. Given a set of hotspots $\mathcal{H}$, the locations in $\mathcal{H}$ are considered preserved if their perturbed locations after rounding are also in $\mathcal{H}$. Formally, the count difference (CD) of a trajectory in hotspot preservation is defined as:

$$CD(\mathcal{T}, \mathcal{T}') = \sum_{i=1}^{|\mathcal{T}|} \mathbf{1}\{\tau_i \in \mathcal{H}\} - \sum_{i=1}^{|\mathcal{T}|} \mathbf{1}\{\tau_i \in \mathcal{H} \wedge \tau'_i \in \mathcal{H}\},$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. A smaller CD indicates better trajectory utility. We can find the relationship between AE and these two metrics. For range query preservation with a given threshold δ , when $AE \leq \delta$, the perturbed location is expected to satisfy the range query. For hotspot preservation a smaller AE leads to better hotspot retention after rounding in TraCS.

We further evaluate the performance of TraCS-D and existing methods under varying privacy parameters ϵ using these two metrics. Specifically, we use the CHI dataset and adopt the following experimental setups:

- Range query preservation: We set the threshold $\delta = 0.1$, and compute the RQP averaged over all trajectories.
- Hotspot preservation: The hotspot set $\mathcal{H}$ is defined as the first 20% frequent locations in the CHI dataset and we compute the CD averaged over all trajectories.

The results are shown in Figure 10. It can be seen that TraCS generally outperforms existing methods across different privacy levels, demonstrating better effectiveness in preserving both range queries and hotspots. Statistically, (i) the mean RQP across all ϵ values for L-SRR, ATP, and NGram are 47.1%, 46.8%, and 56.5%,

respectively, while the mean RQP of TraCS-D and TraCS-C are 58.6% and 68.4%, respectively. (ii) The mean CD across all ϵ values for L-SRR, ATP, and NGram are 5.4, 4.9, and 5.2, respectively, while the mean CD of TraCS-D and TraCS-C are both 4.1. Particularly, TraCS shows improvement in hotspot preservation compared to existing methods across all privacy levels.

4.2.6 Assigning ϵ for a Whole Trajectory. We also evaluate TraCS when the privacy parameter ϵ is specified for an entire trajectory rather than per location. Specifically, for TraCS, L-SRR, NGram, and ATP, we set ϵ at the trajectory level and allocate it uniformly across locations by assigning each location a parameter of $\epsilon/|\mathcal{T}|$, where $|\mathcal{T}|$ is the trajectory length. This uniform allocation is simple and requires no additional information, and we consider it the most robust choice. In contrast, non-uniform allocations (e.g. assigning a larger parameter to utility-critical locations) may improve utility, but they typically rely on public notions of “importance” and can weaken privacy protection for those locations. Appendix C.10 provides results and discussion on this evaluation, showing that TraCS outperforms existing methods in trajectory utility under this setting when ϵ is large.

4.3 Summary of Evaluation

From the evaluation results on both continuous and discrete spaces, we can summarize the following findings:

- In continuous spaces, TraCS generally exhibits better trajectory utility than the strawman method, particularly when the privacy parameter ϵ is large. Meanwhile, TraCS-C has better utility than TraCS-D for common-shape location space and trajectories.
- In discrete spaces, TraCS generally outperforms NGram, L-SRR, and ATP, particularly when the privacy parameter ϵ is large. Furthermore, TraCS has significantly lower time cost, requiring less than 1% of their time cost.

These findings validate the effectiveness and efficiency of TraCS in collecting trajectory data under LDP in both continuous and discrete spaces.

5 Conclusions

This paper presents TraCS-D and TraCS-C, providing local differential privacy guarantees for trajectory collection in continuous location spaces. TraCS-D uses a novel direction-distance perturbation procedure, while TraCS-C perturbs the Cartesian coordinates of each location. These two methods can also be applied to discrete location spaces by rounding each perturbed location to the nearest discrete point, achieving better efficiency than existing discrete methods. Trajectory utility of TraCS is analyzed theoretically and evaluated empirically. Evaluation results on discrete location spaces show that TraCS outperforms state-of-the-art discrete methods in trajectory utility, particularly when the privacy parameter is large.

In summary, TraCS demonstrates that collecting trajectory data in continuous spaces generally better discrete approaches, avoiding privacy and efficiency issues caused by discretization, while can also be rounded to discrete spaces when necessary.

Acknowledgments

We thank the anonymous reviewers and the revision editor for their valuable feedback and guidance, which significantly improved this paper. We also acknowledge the use of GPT-5 for language refinement in this paper. This research is supported in part by the U.S. National Science Foundation under grants CNS-2245689 (CRII), as well as by the 2022 Meta Research Award for Privacy-Enhancing Technologies.

References

- [1] 2016. General Data Protection Regulation (GDPR) - Legal Text. <https://gdpr-info.eu/>
- [2] Miguel E. Andrés, Nicolás Emilio Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: differential privacy for location-based systems. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung (Eds.). ACM, 901–914. <https://doi.org/10.1145/2508859.2516735>
- [3] Erik Buchholz, Alsharif Abuadba, Shuo Wang, Surya Nepal, and Salil S. Kanhere. 2024. SoK: Can Trajectory Generation Combine Privacy and Utility? *Proc. Priv. Enhancing Technol.* 2024, 3 (2024), 75–93. <https://doi.org/10.56553/POPETS-2024-0068>
- [4] Hui Cai, Chen Lan, Biyun Sheng, Jian Zhou, Yuanyuan Yang, Yanmin Zhu, and Fu Xiao. 2026. ADGTrace: Achieving Adaptive Trajectory Synthesis With Generated Data. *IEEE Trans. Mob. Comput.* 25, 1 (2026), 148–163. <https://doi.org/10.1109/TMC.2025.3590770>
- [5] Bo Chen and Matthew T. Hale. 2024. The Bounded Gaussian Mechanism for Differential Privacy. *J. Priv. Confidentiality* 14, 1 (2024). <https://doi.org/10.29012/JPC.850>
- [6] Rui Chen, Gergely Ács, and Claude Castelluccia. 2012. Differentially private sequential data publication via variable-length n-grams. In *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, Ting Yu, George Danezis, and Virgil D. Gligor (Eds.). ACM, 638–649. <https://doi.org/10.1145/2382196.2382263>
- [7] Rui Chen, Benjamin C. M. Fung, and Bipin C. Desai. 2011. Differentially Private Trajectory Data Publication. *CoRR* abs/1112.2020 (2011). arXiv:1112.2020 <http://arxiv.org/abs/1112.2020>
- [8] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, Chid Apté, Joydeep Ghosh, and Padhraic Smyth (Eds.). ACM, 1082–1090. <https://doi.org/10.1145/2020408.2020579>
- [9] Teddy Cunningham, Graham Cormode, and Hakan Ferhatosmanoglu. 2021. Privacy-Preserving Synthetic Location Data in the Real World. In *Proceedings of the 17th International Symposium on Spatial and Temporal Databases, SSTD 2021, Virtual Event, USA, August 23-25, 2021*, Erik Hoel, Dev Oliver, Raymond Chi-Wing Wong, and Ahmed Eldawy (Eds.). ACM, 23–33. <https://doi.org/10.1145/3469830.3470893>
- [10] Teddy Cunningham, Graham Cormode, Hakan Ferhatosmanoglu, and Divesh Srivastava. 2021. Real-World Trajectory Sharing with Local Differential Privacy. *Proc. VLDB Endow.* 14, 11 (2021), 2283–2295. <https://doi.org/10.14778/3476249.3476280>
- [11] Yuntao Du, Yujia Hu, Zhikun Zhang, Ziquan Fang, Lu Chen, Baihua Zheng, and Yunjun Gao. 2023. LDPTrace: Locally Differentially Private Trajectory Synthesis. *Proc. VLDB Endow.* 16, 8 (2023), 1897–1909. <https://doi.org/10.14778/3594512.3594520>
- [12] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. 2013. Local Privacy and Statistical Minimax Rates. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*. IEEE Computer Society, 429–438. <https://doi.org/10.1109/FOCS.2013.53>
- [13] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. 2013. Local Privacy, Data Processing Inequalities, and Statistical Minimax Rates. *arXiv: Statistics Theory* (2013). <https://api.semanticscholar.org/CorpusID:18651360>
- [14] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 4052)*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (Eds.). Springer, 1–12. https://doi.org/10.1007/11787006_1
- [15] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* 9, 3-4 (2014), 211–407. <https://doi.org/10.1561/04000000042>
- [16] Xiaolan Gu, Ming Li, Li Xiong, and Yang Cao. 2020. Providing Input-Discriminative Protection for Local Differential Privacy. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*. IEEE, 505–516. <https://doi.org/10.1109/ICDE48307.2020.00050>
- [17] Naoise Holohan, Spiros Antonatos, Stefano Braghin, and Pól Mac Aonghusa. 2020. The Bounded Laplace Mechanism in Differential Privacy. *J. Priv. Confidentiality* 10, 1 (2020). <https://doi.org/10.29012/JPC.715>
- [18] I-Jung Hsu, Chih-Hsun Lin, Chia-Mu Yu, Sy-Yen Kuo, and Chun-Ying Huang. 2025. Poisoning Attacks to Local Differential Privacy Protocols for Trajectory Data. *CoRR* abs/2503.07483 (2025). <https://doi.org/10.48550/ARXIV.2503.07483>
- [19] Hongbo Jiang, Jie Li, Ping Zhao, Fanzi Zeng, Zhu Xiao, and Arun Iyengar. 2022. Location Privacy-preserving Mechanisms in Location-based Services: A Comprehensive Survey. *ACM Comput. Surv.* 54, 1 (2022), 4:1–4:36. <https://doi.org/10.1145/3423165>
- [20] Kaifeng Jiang, Dongxu Shao, Stéphane Bressan, Thomas Kister, and Kian-Lee Tan. 2013. Publishing trajectories with differential privacy guarantees. In *Conference on Scientific and Statistical Database Management, SSDBM '13, Baltimore, MD, USA, July 29 - 31, 2013*, Alex Szalay, Tamas Budavari, Magdalena Balazinska, Alexandra Meliou, and Ahmet Sacan (Eds.). ACM, 12:1–12:12. <https://doi.org/10.1145/2484838.2484846>
- [21] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2014. Extremal Mechanisms for Local Differential Privacy. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 2879–2887. <https://proceedings.neurips.cc/paper/2014/hash/86df7dcfd896fc2674f757a2463eba-Abstract.html>
- [22] John Krumm. 2009. A survey of computational location privacy. *Pers. Ubiquitous Comput.* 13, 6 (2009), 391–399. <https://doi.org/10.1007/S00779-008-0212-5>
- [23] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20, 2007*, Rada Chirkova, Asuman Dogac, M. Tamer Özsu, and Timos K. Sellis (Eds.). IEEE Computer Society, 106–115. <https://doi.org/10.1109/ICDE.2007.367856>
- [24] Yan-zi Li, Li Xu, Jing Zhang, and Liao-ru-xing Zhang. 2025. WF-LDPSR: A local differential privacy mechanism based on water-filling for secure release of trajectory statistics data. *Comput. Secur.* 148 (2025), 104165. <https://doi.org/10.1016/J.COSE.2024.104165>
- [25] Zitao Li, Tianhao Wang, Milan Lopuhaa-Zwakenberg, Ninghui Li, and Boris Skoric. 2020. Estimating Numerical Distributions under Local Differential Privacy. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo (Eds.). ACM, 621–635. <https://doi.org/10.1145/3318464.3389700>
- [26] Fang Liu. 2019. Statistical Properties of Sanitized Results from Differentially Private Laplace Mechanism with Univariate Bounding Constraints. *Trans. Data Priv.* 12, 3 (2019), 169–195. <http://www.tdp.cat/issues16/tdp.a316a18.pdf>
- [27] Xinyi Liu, Ye Zheng, Zhengxiong Li, and Yidan Hu. 2024. Multi-sensor Data Privacy Protection with Adaptive Privacy Budget for IoT Systems. In *2024 IEEE Conference on Communications and Network Security (CNS)*, 1–9. <https://doi.org/10.1109/CNS62487.2024.10735696>
- [28] Fei Ma, Renbo Zhu, and Ping Wang. 2024. PTT: Piecewise Transformation Technique for Analyzing Numerical Data Under Local Differential Privacy. *IEEE Trans. Mob. Comput.* 23, 10 (2024), 9518–9531. <https://doi.org/10.1109/TMC.2024.3364496>
- [29] Ricardo Mendes, Mariana Cunha, and João P. Vilela. 2020. Impact of Frequency of Location Reports on the Privacy Level of Geo-indistinguishability. *Proc. Priv. Enhancing Technol.* 2020, 2 (2020), 379–396. <https://doi.org/10.2478/POPETS-2020-0032>
- [30] Alex Miranda-Pascual, Patricia Guerra-Balboa, Javier Parra-Arnau, Jordi Forné, and Thorsten Strufe. 2023. SoK: Differentially Private Publication of Trajectory Data. *Proc. Priv. Enhancing Technol.* 2023, 2 (2023), 496–516. <https://doi.org/10.56553/POPETS-2023-0065>
- [31] Kobbi Nissim, Rann Smorodinsky, and Moshe Tennenholtz. 2012. Approximately optimal mechanism design via differential privacy. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, Shafi Goldwasser (Ed.). ACM, 203–213. <https://doi.org/10.1145/2090236.2090254>
- [32] Chenxi Qiu, Ruiyao Liu, Primal Pappachan, Anna Cinzia Squicciarini, and Xinpeng Xie. 2025. Time-Efficient Locally Relevant Geo-Location Privacy Protection. *Proc. Priv. Enhancing Technol.* 2025, 2 (2025), 5–22. <https://doi.org/10.56553/POPETS-2025-0046>
- [33] Latanya Sweeney. 2002. k-Anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* 10, 5 (2002), 557–570. <https://doi.org/10.1142/S0218488502001648>
- [34] Montana State University. n.d.. Lat/Lon and UTM Conversion - Yellowstone Research Coordination Network. <https://rcn.montana.edu/resources/Converter.aspx>
- [35] Han Wang, Hanbin Hong, Li Xiong, Zhan Qin, and Yuan Hong. 2022. L-SRR: Local Differential Privacy for Location-Based Services with Staircase Randomized

- Response. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi (Eds.). ACM, 2809–2823. <https://doi.org/10.1145/3548606.3560636>
- [36] Nana Wang and Mohan S. Kankanhalli. 2020. Protecting sensitive place visits in privacy-preserving trajectory publishing. *Comput. Secur.* 97 (2020), 101949. <https://doi.org/10.1016/j.cose.2020.101949>
- [37] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. 2019. Collecting and Analyzing Multidimensional Data with Local Differential Privacy. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*. IEEE, 638–649. <https://doi.org/10.1109/ICDE.2019.00063>
- [38] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally Differentially Private Protocols for Frequency Estimation. In *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*, Engin Kirda and Thomas Ristenpart (Eds.). USENIX Association, 729–745. <https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/wang-tianhao>
- [39] Benjamin Weggenmann and Florian Kerschbaum. 2021. Differential Privacy for Directional Data. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, Yongdae Kim, Jong Kim, Giovanni Vigna, and Elaine Shi (Eds.). ACM, 1205–1222. <https://doi.org/10.1145/3460120.3484734>
- [40] Dingqi Yang, Daqing Zhang, Vincent W. Zheng, and Zhiyong Yu. 2015. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. *IEEE Trans. Syst. Man Cybern. Syst.* 45, 1 (2015), 129–142. <https://doi.org/10.1109/TSMC.2014.2327053>
- [41] Haolong Yang, Dingyuan Shi, Yuanyuan Zhang, Yi Xu, and Ke Xu. 2024. An Efficient Local Differential Privacy Approach for Trajectory Publishing with High Utility. In *Database Systems for Advanced Applications, Makoto Onizuka, Jae-Gil Lee, Yongxin Tong, Chuan Xiao, Yoshiharu Ishikawa, Sihem Amer-Yahia, H. V. Jagadish, and Kejun Lu (Eds.)*. Springer Nature Singapore, Singapore, 71–88.
- [42] Dongyue Zhang, Weiwei Ni, Nan Fu, Lihe Hou, and Ruyi Zhang. 2025. Locally Differentially Private Trajectory Publication Based on Regional Popularity Awareness. *IEEE Trans. Inf. Forensics Secur.* 20 (2025), 6719–6732. <https://doi.org/10.1109/TIFS.2025.3577967>
- [43] Jun Zhang, Xiaokui Xiao, and Xing Xie. 2016. PrivTree: A Differentially Private Algorithm for Hierarchical Decompositions. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, Fatma Özcan, Georgia Koutrika, and Sam Madden (Eds.). ACM, 155–170. <https://doi.org/10.1145/2882903.2882928>
- [44] Yuemin Zhang, Qingqing Ye, Rui Chen, Haibo Hu, and Qilong Han. 2023. Trajectory Data Collection with Local Differential Privacy. *Proc. VLDB Endow.* 16, 10 (2023), 2591–2604. <https://doi.org/10.14778/3603581.3603597>
- [45] Xiaodong Zhao, Dechang Pi, and Junfu Chen. 2020. Novel trajectory privacy-preserving method based on clustering using differential privacy. *Expert Syst. Appl.* 149 (2020), 113241. <https://doi.org/10.1016/j.eswa.2020.113241>
- [46] Yang Zhao, Jun Zhao, Mengmeng Yang, Teng Wang, Ning Wang, Lingjuan Lyu, Dusit Niyato, and Kwok-Yan Lam. 2021. Local Differential Privacy-Based Federated Learning for Internet of Things. *IEEE Internet Things J.* 8, 11 (2021), 8836–8853. <https://doi.org/10.1109/JIOT.2020.3037194>
- [47] Ye Zheng, Sumita Mishra, and Yidan Hu. 2025. Optimal Piecewise-based Mechanism for Collecting Bounded Numerical Data under Local Differential Privacy. *Proc. Priv. Enhancing Technol.* 2025, 4 (2025), 146–165. <https://doi.org/10.56553/POPETS-2025-0124>

A Related Work

This section reviews related trajectory collection methods under LDP and other quantifiable privacy notions. More privacy-preserving trajectory collection methods can be found in recent surveys [3, 30].

Trajectory Collection under LDP. We have discussed three state-of-the-art trajectory collection methods under pure LDP: NGram [10], L-SRR [35], and ATP [44]. Among them, L-SRR was originally applied for origin-destination location pairs, while NGram, ATP, and this paper focus on trajectory perturbation. NGram attempts to ensure pure LDP for the “trajectory space”, also named as trajectory-level or instance-level LDP in surveys [3, 30], which is a stronger privacy than the location space. However, this is proved extremely challenging as the trajectory space increases exponentially with the number of possible locations in the location space. L-SRR, ATP and

this paper focus on ensuring pure LDP for the location space. Unlike L-SRR, ATP and other existing methods based on discrete mechanisms, this paper focuses on continuous location spaces, which are more expressive than any fine-grained discrete location space.

Beyond the above three methods that design LDP mechanisms for trajectory collection, several works leverage external knowledge to improve the utility of LDP-based trajectory collection. WFLDPSR [24] adaptively determines each user’s privacy protection level based on the sensitivity of the user’s information, and applies a water-filling strategy to allocate privacy budgets during perturbation. Regional popularity (i.e. the popularity of different regions in the location space) can also be exploited to constrain trajectory boundaries and improve utility [42]. In addition, recent work studies poisoning attacks against NGram and ATP [18], where adversaries induce malicious users to submit poisoned trajectories to the data collector with the goal of promoting target patterns.

Another line of research is trajectory *synthesis* for privacy-preserving trajectory publication. A typical work is LDPTrace [11], which synthesizes trajectories using a generative model. LDPTrace discretizes the location space into cells and represents each trajectory as a sequence of cells. Transition patterns between neighboring cells are perturbed using LDP mechanisms and collected by an untrusted curator. The curator then trains a Markov chain model on the perturbed transition patterns to capture movement behaviors. Following LDPTrace, ADGTrace [4] also employs Markov chain models for trajectory synthesis; it does not provide LDP guarantees and leverages personal features to improve utility. Compared to methods that perturb each trajectory, such as NGram, L-SRR, and ATP, synthetic trajectories may be overly too random and not reflect the original trajectories if not personalized enough.

LDP Mechanisms for Bounded Numerical Domains. The perturbation mechanisms used in TraCS for direction and distance perturbation are utility-optimized piecewise-based mechanisms in OGPM [47]. Other LDP mechanisms for bounded numerical domains [17, 26, 39] can also be incorporated into TraCS; refer to Section 3.5.6 for discussions and comparisons. Compared with OGPM, which focuses on optimizing the 1D mechanisms’ data utility, the design of 2D mechanisms in TraCS needs to consider the unique characteristics of trajectory data and 2D spaces. In particular, the direction-distance perturbation in TraCS-D to guarantee LDP for a rectangular region requires careful design. Meanwhile, this paper emphasizes the overlooked fundamental issues inherent in discrete-space LDP mechanisms for trajectory collection [10, 35, 44], which make privacy, utility, and efficiency discretization-dependent and prevent guarantees for continuous spaces. TraCS demonstrates that designing mechanisms directly in continuous space (via building blocks like OGPM, etc.) avoids these issues and subsumes discrete cases via rounding.

Trajectory Collection under DP. As a weaker privacy notion than LDP, (central) DP [14, 15] assumes a trusted curator who processes the raw data and releases the perturbed data. It requires that only neighboring locations in the location space be indistinguishable, rather than any two arbitrary locations. Compared to LDP, the concept of “neighboring” in DP has many interpretations. A comprehensive survey on DP-based trajectory collection and publication is available in [30].

The use of direction information in trajectory collection dates back to an early work SDD [20], which attempts to add unbounded Laplace noise to the direction information. It provides a DP guarantee for unbounded continuous location spaces, but results in poor utility. The notion of “n-gram” in trajectory collection dates back to [6], where it abstracts a trajectory pattern as a gram. The idea of hierarchical decomposition can also be found in [43], where a spatial decomposition method is used for trajectory synthesis.

External knowledge is also widely used in DP-based trajectory collection methods. NGram’s prior work [9] provides a DP-based solution for trajectory synthesis by exploiting the public knowledge of the road network. Focusing on particular privacy-concerned locations, or sensitive zones [36], can also improve the trajectory utility while maintaining partial privacy guarantees.

Trajectory Collection under Other Privacy Notions. A widely used relaxed privacy notion than LDP for trajectory collection is Geo-indistinguishability [2, 29], where the indistinguishability between two locations is a function of their distance, i.e. privacy can decrease with distance. Input-discriminative LDP [16] goes one step further, assigning each location its own privacy level. Threshold-integrated LDP [41] defines a threshold to constrain the sensitive region near a location. This is similar to NGram’s reachability constraint but with a formal privacy definition. More relaxed LDP notions for trajectory collection can be found in paper survey [30]. Although these relaxed LDP notions generally provide better trajectory utility than pure LDP, they sacrifice the strong privacy guarantee of pure LDP, and often need hyperparameters to define the privacy level.

B Proofs

B.1 LDP Proof of Definition 5

The proof applies to all piecewise-based mechanisms, including the direction perturbation mechanism in Definition 5 and the distance perturbation mechanism in Definition 6.

PROOF. For any two sensitive directions φ_1, φ_2 and any $\varphi' \in [0, 2\pi)$, we have:

$$\frac{pdf[\mathcal{M}_o(\varphi_1) = \varphi']}{pdf[\mathcal{M}_o(\varphi_2) = \varphi']} \leq \frac{p_\varepsilon}{p_\varepsilon/\exp(\varepsilon)} = \exp(\varepsilon).$$

The inequality holds because any φ' is sampled with a probability of at most p_ε and at least $p_\varepsilon/\exp(\varepsilon)$, regardless of whether the input is φ_1, φ_2 or any other value. $\square$

B.2 Proof of Theorem 2

PROOF. Denote TraCS-D in Algorithm 1 as $\mathcal{M}$. We need to prove that $\mathcal{M}$ satisfies ε -LDP for each location $\tau \in \mathcal{S}$, i.e.

$$\forall \tau_1, \tau_2, \tau' \in \mathcal{S} : \frac{pdf[\mathcal{M}(\tau_1) = \tau']}{pdf[\mathcal{M}(\tau_2) = \tau']} \leq \exp(\varepsilon).$$

In TraCS-D, each location τ is represented in a direction-distance space with unique coordinates $(\varphi, r(\varphi))$. The key insight of the proof is that the perturbation of φ and $r(\varphi)$ satisfies ε_d -LDP and $(\varepsilon - \varepsilon_d)$ -LDP, respectively. Then TraCS-D ensures ε -LDP for the location space $\mathcal{S} = \mathcal{D}_\varphi \otimes \mathcal{D}_{r(\varphi)}$. We provide proofs by the Composition Theorem 1 and by computing the 2D pdf ratios in the LDP definition.

By Composition Theorem 1. The definitions of piecewise-based mechanisms $\mathcal{M}_o$ and $\mathcal{M}_-$ imply that they satisfy LDP. Specifically, for $\mathcal{M}_o(\varphi; \varepsilon_d)$,

$$\forall \varphi_1, \varphi_2, \varphi' \in [0, 2\pi) : \frac{pdf[\mathcal{M}_o(\varphi_1) = \varphi']}{pdf[\mathcal{M}_o(\varphi_2) = \varphi']} \leq \exp(\varepsilon_d),$$

is guaranteed by Definition 5 of $\mathcal{M}_o$. Similarly, $\mathcal{M}_-(\bar{r}(\varphi); \varepsilon - \varepsilon_d)$ in Definition 6 ensures $(\varepsilon - \varepsilon_d)$ -LDP for $\bar{r}(\varphi)$. For clarity, let $\bar{r} := \bar{r}(\varphi)$, which yields

$$\forall \bar{r}_1, \bar{r}_2, \bar{r}' \in [0, 1) : \frac{pdf[\mathcal{M}_-(\bar{r}_1) = \bar{r}']}{pdf[\mathcal{M}_-(\bar{r}_2) = \bar{r}']} \leq \exp(\varepsilon - \varepsilon_d).$$

The subsequent linear mapping from $\bar{r} \in [0, 1)$ to $r \in \mathcal{D}_{r(\varphi')}$ is a post-processing step that does not affect the LDP property. Specifically, denote the linear mapping as $g : [0, 1) \rightarrow \mathcal{D}_{r(\varphi')}$, and denote $r_1 = g(\mathcal{M}_-(\bar{r}_1))$ and $r_2 = g(\mathcal{M}_-(\bar{r}_2))$ as two random variables. It is clear that

$$\forall r_1, r_2, r' \in \mathcal{D}_{r(\varphi')} : \frac{pdf[r_1 = r']}{pdf[r_2 = r']} \leq \exp(\varepsilon - \varepsilon_d),$$

because the linear mapping g is a deterministic function without randomness. Thus, $\mathcal{M}_-$ ensures $(\varepsilon - \varepsilon_d)$ -LDP for $\mathcal{D}_{r(\varphi')}$.

Combining these results with Sequential Composition Theorem 1, we can conclude that $\mathcal{M} = (\mathcal{M}_o, \mathcal{M}_-)$ ensures ε -LDP for each location $\tau = (\varphi, r(\varphi)) \in \mathcal{S}$. Consequently, the entire trajectory with n locations satisfies $n\varepsilon$ -LDP.

By computing the 2D pdf ratio. Assuming two sensitive locations $\tau_1 = (\varphi_1, \bar{r}_1)$ and $\tau_2 = (\varphi_2, \bar{r}_2)$, where $\bar{r}_1$ and $\bar{r}_2$ are normalized distances from the reference location along φ_1 and φ_2 , respectively. For any output $\tau' = (\varphi', \bar{r}')$, the 2D pdf of getting it from τ_1 and τ_2 are

$$\begin{aligned} pdf[\mathcal{M}(\tau_1) = \tau'] &= pdf[\mathcal{M}_o(\varphi_1) = \varphi'] \cdot pdf[\mathcal{M}_-(\bar{r}_1) = \bar{r}'], \\ pdf[\mathcal{M}(\tau_2) = \tau'] &= pdf[\mathcal{M}_o(\varphi_2) = \varphi'] \cdot pdf[\mathcal{M}_-(\bar{r}_2) = \bar{r}'], \end{aligned}$$

where $\mathcal{M}_-$ is applied to the *same* normalized distance space (from the reference location to the boundary of $\mathcal{S}$ along φ'). The ratio is bounded by $\exp(\varepsilon_d) \cdot \exp(\varepsilon - \varepsilon_d) = \exp(\varepsilon)$ due to the definition of $\mathcal{M}_o$ and $\mathcal{M}_-$. Consequently, the entire trajectory with n locations satisfies $n\varepsilon$ -LDP. $\square$

B.3 Proof of Theorem 3

PROOF. We prove the $\Theta(e^{-\varepsilon/2})$ convergence rate by calculating the MSE of the perturbation mechanism. Specifically, the MSE of $\mathcal{M}_-$ is calculated as follows:

$$\text{MSE}[\mathcal{M}_-(\bar{r})] = \int_0^1 (\bar{r}^* - \bar{r})^2 pdf[\mathcal{M}_-(\bar{r}) = \bar{r}^*] d\bar{r}^*.$$

The worst-case MSE is achieved when $\bar{r} = 0$ or $\bar{r} = 1$, i.e. at the endpoints (similar to the variance calculation in [37]). In this case,

$$\begin{aligned} \text{MSE}[\mathcal{M}_-(0)] &= \int_0^{2C} (\bar{r}^* - 0)^2 p_\varepsilon d\bar{r}^* + \int_{2C}^1 (\bar{r}^* - 0)^2 \frac{p_\varepsilon}{e^\varepsilon} d\bar{r}^* \\ &= \frac{8C^3 p_\varepsilon}{3} + \frac{(1 - 8C^3) p_\varepsilon}{3e^\varepsilon}. \end{aligned}$$

with p_ε and C defined in Definition 6. We omit the calculation for $\bar{r} = 1$ as it is symmetric to $\bar{r} = 0$. Plugging in the values of p_ε and C ,

we can simplify the MSE using big O notation:

$$\begin{aligned} \text{MSE}[\mathcal{M}_-(0)] &= \frac{1}{3}e^{-\varepsilon/2} + \frac{(e^{\varepsilon/2} - 1)^3}{3e^{\varepsilon/2}(e^\varepsilon - 1)^2} \\ &= \frac{1}{3}e^{-\varepsilon/2} + O(e^{-\varepsilon}). \end{aligned}$$

Therefore, as $\varepsilon \rightarrow \infty$, the worst-case MSE of $\mathcal{M}_-$ converges to zero at a rate of $\Theta(e^{-\varepsilon/2})$. The MSE of $\mathcal{M}_0$ can be calculated in the same way, resulting in a convergence rate of $\Theta(e^{-\varepsilon/2})$ as well. $\square$

B.4 Proof of Theorem 4

PROOF. Denote TraCS-C in Algorithm 2 as $\mathcal{M}$. We show that $\mathcal{M}$ satisfies ε -LDP for each location $\tau \in \mathcal{S}$. In TraCS-C, each location τ is represented in the Cartesian space $\mathcal{D}_a \times \mathcal{D}_b$ with unique coordinates (a, b) . Thus, it suffices to prove that the perturbations of a and b each satisfy $\varepsilon/2$ -LDP.

From Theorem 2, $\mathcal{M}_-(\bar{a}; \varepsilon/2)$ satisfies $\varepsilon/2$ -LDP for the normalized coordinate $\bar{a} \in [0, 1)$. The subsequent linear mapping from $\bar{a} \in [0, 1)$ to $a \in \mathcal{D}_a$ is post-processing and therefore preserves the LDP guarantee. Hence, $\mathcal{M}_-$ ensures $\varepsilon/2$ -LDP over $\mathcal{D}_a$. By the same argument, $\mathcal{M}_-$ also ensures $\varepsilon/2$ -LDP over $\mathcal{D}_b$.

Therefore, $\mathcal{M} := (\mathcal{M}_-, \mathcal{M}_-)$ satisfies ε -LDP for each location $\tau = (a, b) \in \mathcal{S}$ by the Sequential Composition Theorem 1. Equivalently, one can verify the guarantee by directly bounding the 2D *pdf* ratio induced by $\mathcal{M}$. Consequently, the entire trajectory with n locations satisfies $n\varepsilon$ -LDP. $\square$

C Complementary Materials

C.1 Space-Dependence of Indistinguishability (Section 1)

Indistinguishability of a data point is always relative to a specified data space, which determines the set of alternatives that an adversary may try to distinguish it from. Accordingly, indistinguishability cannot be meaningfully discussed without first specifying the underlying space, for two main reasons.

(i) LDP perspective. The LDP guarantee is defined with respect to a particular input domain. If two algorithms operate on the same domain and use the same privacy parameter ε , then their privacy guarantees are comparable (in the sense of the LDP definition); if their domains differ, then their guarantees may not be directly comparable, even when they share the same ε .

(ii) Information-theoretic perspective. For an ε -LDP mechanism $\mathcal{M}$, the mutual information $I(x; \mathcal{M}(x))$ is upper bounded by a function of both ε and the size of the input space, i.e. $I(x; \mathcal{M}(x)) \leq O(\varepsilon, |\mathcal{X}|)$, where $|\mathcal{X}|$ denotes the cardinality of the input space $\mathcal{X}$. A similar bound can be found in [13].*

This observation is particularly relevant to trajectory collection or synthesis under LDP when discretization-based methods are used. Even for a fixed continuous area, different discretization strategies (e.g. uniform grids, adaptive grids, or different sets of points of interest (POIs)) induce different discrete location spaces with different sizes and spatial layouts. Consequently, the effective

*Intuitively, LDP enforces *pairwise* indistinguishability but does not directly account for how indistinguishability accumulates over many alternatives, which can be captured by mutual information.

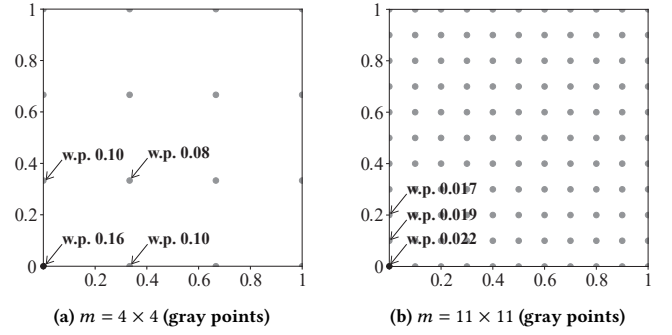


Figure 11: The Exponential mechanism $\mathcal{M}_{\text{exp}}(0, 0)$ with $\varepsilon = 4$ on two discrete location spaces within $[0, 1] \times [0, 1]$. The number and arrangement of points affect the mechanism.

indistinguishability provided by the same LDP mechanism can vary substantially across discretizations, even when the same ε is used.

C.2 Limitations of the Exponential Mechanism (Section 1 and Section 2.3)

C.2.1 High Computational Complexity. The biggest limitation of the Exponential mechanism is the complexity in calculating the score function and sampling from the probability distribution. Before perturbing location x , the Exponential mechanism requires computing the score function $d(x, y)$ for every possible location y in the location space. If the location space has m locations, the time complexity of computing the score function for a single location is $O(m)$, which is computationally expensive for large-scale spaces. Moreover, ATP's dynamic strategy to constrain the location space for perturbation makes it need to recompute the score function for each location, resulting in $O(m)$ time complexity, where m is the size of the location space.

The complexity of sampling from the Exponential mechanism is also $O(m)$, as its cumulative distribution function (CDF) is an m -piece function. Sampling requires comparing a random number with the CDF values of all m locations.

In contrast, we have seen that piecewise-based perturbation mechanisms in TraCS do not require score functions, and they sample from a 3-piece CDF at each perturbation, resulting in a negligible constant time complexity, i.e. $\Theta(1)$, for each perturbation.

C.2.2 Affected by the Number and Arrangement of Locations. Because the sampling probability depends on the score function, the spatial distribution and relative distances between locations further affect the probability assigned to each location.

Figure 11 illustrates the Exponential mechanism $\mathcal{M}_{\text{exp}}$ applied to the sensitive location $\tau = (0, 0)$ with $\varepsilon = 4$ on two discrete location spaces within $[0, 1] \times [0, 1]$. Specifically, Figure 11a presents a 4×4 grid of locations, while Figure 11b depicts an 11×11 grid. We can observe that the sampling probabilities of $\mathcal{M}_{\text{exp}}$ differ for the same sensitive location $\tau = (0, 0)$, even though both location spaces are contained within the same $[0, 1] \times [0, 1]$ area. When the number of locations increases and the arrangement becomes denser (i.e. $m = 11 \times 11$), the sampling probabilities for each location become smaller and more uniform.

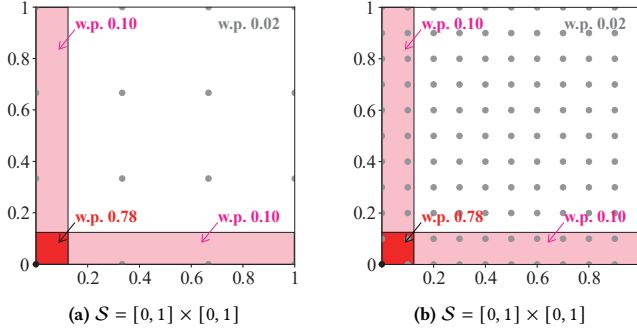


Figure 12: TraCS-C mechanism $\mathcal{M}_{\text{TraCS-C}}(0,0)$ with $\epsilon = 4$ on $S = [0, 1] \times [0, 1]$, which encompasses both discrete location spaces in Figure 11. The sampling probability is defined over areas rather than individual points.

In contrast, Figure 12 illustrates the TraCS-C mechanism $\mathcal{M}_{\text{TraCS-C}}$ applied to the same sensitive location $\tau = (0, 0)$ with $\epsilon = 4$ over the $[0, 1] \times [0, 1]$ area. Unlike the Exponential mechanism, TraCS-C is designed for continuous spaces, where the sampling probability is defined over areas rather than individual points.

To apply TraCS-C to the discrete spaces shown in Figure 11, one can simply round each perturbed location to its nearest discrete point. Since this rounding is a post-processing step, it does not affect the LDP guarantee.

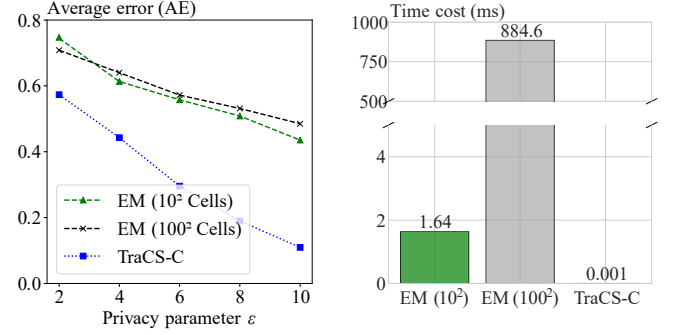
C.3 Limitations of Applying Discrete LDP Mechanisms to Continuous Spaces (Section 1)

Although a continuous space can be discretized before applying discrete LDP mechanisms, this approach inherits the privacy–utility–efficiency issues of discrete methods and introduces additional challenges in choosing an appropriate discretization strategy.

Gridding the continuous space. A common discretization approach is to partition the continuous space into uniform grid cells and treat each cell as a discrete input and output of the discrete mechanism. The reported cell can then be post-processed by randomly sampling a location within that cell.

Dilemma. However, the gridding approach faces a fundamental dilemma in choosing the grid cell size, which directly affects the privacy–utility–efficiency tradeoff. (i) If the grid cells are too large (i.e. coarse-grained gridding), the mechanism is more likely to output the true cell, but the intra-cell error (i.e. the distance between the true location and the post-processed location sampled within the same cell) can be large. (ii) If the grid cells are too small (i.e. fine-grained gridding), the intra-cell error can be reduced, but the probability of outputting the true cell decreases. Meanwhile, the computational cost increases with the number of cells.

We empirically evaluate the gridding approach under different cell sizes in a continuous space and compare it with TraCS-C. Specifically, we consider the $[0, 1] \times [0, 1]$ location space and two grid resolutions: 10×10 and 100×100 . For a fixed location, we generate 500 perturbed locations using (i) the gridding approach with the Exponential mechanism (EM), where the score is defined by the distance between cell centers, and (ii) TraCS-C. We then report the



(a) Error of the gridding approach with different cell sizes. (b) Average time cost of perturbing one location.

Figure 13: Performance of the gridding approach with different cell sizes in a continuous space. Coarse-grained grids (e.g. 10×10) perform better when ϵ is small, whereas fine-grained grids (e.g. 100×100) perform better when ϵ is large, at the cost of higher computational time. In contrast, TraCS achieves better utility than both gridding baselines across all ϵ values, with negligible time overhead.

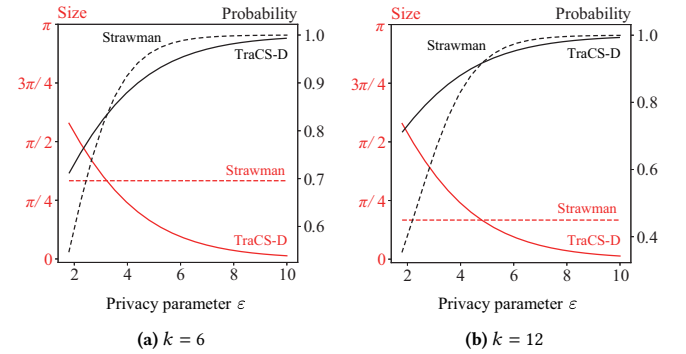


Figure 14: Comparison of the size and probability of the dominant sector between TraCS-D and the strawman approach. TraCS-D (solid line) achieves a better trade-off between the size and probability of the dominant sector compared to the strawman approach (dashed line), having better adaptiveness across different ϵ regions.

average error and the average time required to perturb a single location. Figure 13 summarizes the results. In the small- ϵ regime, coarse-grained grids (e.g. 10×10) outperform fine-grained grids (e.g. 100×100), whereas in the large- ϵ regime, fine-grained grids perform better. This trend is consistent with the above dilemma: it is difficult to select a single grid resolution that performs well across different ϵ values. In contrast, TraCS-C consistently achieves better utility than both gridding baselines for all tested ϵ values, while incurring negligible time overhead.

C.4 Dominant Sector Comparison Between the Strawman Approach and TraCS-D (Example 1)

For direction perturbation, a small-size and high-probability dominant sector is desirable for more accurate perturbation. However, these two properties are generally conflicting with each other due

to the LDP constraint: a smaller dominant sector typically has a lower probability. We will show that TraCS-D achieves a better trade-off between the size and probability of the dominant sector compared to the strawman approach with any k value. Figure 14 shows two exemplary quantitative comparisons between them.

Small ϵ region. In this case, the dominant sector of TraCS-D (red solid line) is larger than the strawman approach, but the probability of the dominant sector (black solid line) is significantly higher, especially when k is large. This indicates that, although the dominant sector of TraCS-D is larger in this region, it is sampled with a higher probability.

Large ϵ region. In such cases, the dominant sector of TraCS-D is significantly smaller than the strawman approach, while maintaining almost the same probability as the strawman approach.

We can observe the effect of k on the strawman approach. Larger k values in the strawman approach reduce the inner-sector error compared to TraCS-D, but also significantly decrease the probability of the dominant sector when ϵ is small. Conversely, smaller k values increase the probability of the dominant sector when ϵ is small, but result in a large inner-sector error when ϵ is large. In contrast, TraCS-D adaptively balances these trade-offs by adjusting the dominant sector's size and probability according to ϵ .

C.5 Detailed Form of $|\mathcal{D}_{r(\varphi)}|$ (Section 3.2.2)

For each location τ_i , denote its directions to the four endpoints of the rectangular location space as $\varphi_1, \varphi_2, \varphi_3, \varphi_4$. We have:

$$\begin{aligned}\varphi_1 &= \text{atan2}(b_{\text{end}} - b_i, a_{\text{end}} - a_i), \\ \varphi_2 &= \text{atan2}(b_{\text{end}} - b_i, a_{\text{sta}} - a_i), \\ \varphi_3 &= \text{atan2}(b_{\text{sta}} - b_i, a_{\text{sta}} - a_i) + 2\pi, \\ \varphi_4 &= \text{atan2}(b_{\text{sta}} - b_i, a_{\text{end}} - a_i) + 2\pi.\end{aligned}$$

Then, $|\mathcal{D}_{r(\varphi)}|$ can be derived using trigonometric functions, which leads to the following four cases:

$$|\mathcal{D}_{r(\varphi)}| = \begin{cases} \frac{a_{\text{end}} - a_i}{\cos \varphi} & \text{if } \varphi \in [0, \varphi_1) \cup [\varphi_4, 2\pi), \\ \frac{b_{\text{end}} - b_i}{\sin \varphi} & \text{if } \varphi \in [\varphi_1, \varphi_2), \\ \frac{a_i - a_{\text{sta}}}{-\cos \varphi} & \text{if } \varphi \in [\varphi_2, \varphi_3), \\ \frac{b_i - b_{\text{sta}}}{-\sin \varphi} & \text{if } \varphi \in [\varphi_3, \varphi_4), \end{cases}$$

depending on the direction φ from reference location $\tau_i = (a_i, b_i)$ to the boundary of the rectangular location space $[a_{\text{sta}}, a_{\text{end}}] \times [b_{\text{sta}}, b_{\text{end}}]$.

C.6 Redesign SW (Section 3.5.7)

We redesign the SW mechanism to achieve ϵ -LDP for the circular space $[0, 2\pi)$ and the linear space $[0, 1)$. The key idea is to transform the original sampling distribution to these spaces while preserving the LDP constraint. Similar to Definition 5, the redesigned SW mechanism for direction perturbation is defined as follows.

DEFINITION 7 (REDESIGNED SW FOR DIRECTION PERTURBATION). *Given a sensitive direction φ and a privacy parameter ϵ , redesigned SW for direction perturbation is a mechanism $\mathcal{M}_o : [0, 2\pi) \rightarrow [0, 2\pi)$*

defined by:

$$\text{pdf}[\mathcal{M}_o(\varphi) = \varphi'] = \begin{cases} p_\epsilon & \text{if } \varphi' \in [l_{\varphi, \epsilon}, r_{\varphi, \epsilon}), \\ p_\epsilon / \exp(\epsilon) & \text{otherwise,} \end{cases}$$

where $p_\epsilon = \frac{1}{2\pi\epsilon}(\exp(\epsilon) - 1)$ is the sampling probability, and $[l_{\varphi, \epsilon}, r_{\varphi, \epsilon})$ is the sampling interval that

$$\begin{aligned}l_{\varphi, \epsilon} &= \left(\varphi - \pi \frac{\exp(\epsilon)(\epsilon - 1) + 1}{(\exp(\epsilon) - 1)^2} \right) \bmod 2\pi, \\ r_{\varphi, \epsilon} &= \left(\varphi + \pi \frac{\exp(\epsilon)(\epsilon - 1) + 1}{(\exp(\epsilon) - 1)^2} \right) \bmod 2\pi.\end{aligned}$$

Similar to Definition 6, the redesigned SW mechanism for distance perturbation is defined as follows.

DEFINITION 8 (REDESIGNED SW FOR DISTANCE PERTURBATION). *Given a sensitive distance $\bar{r}(\varphi)$ and a privacy parameter ϵ , redesigned SW for distance perturbation is a mechanism $\mathcal{M}_- : [0, 1) \rightarrow [0, 1)$ that*

$$\text{pdf}[\mathcal{M}_-(\bar{r}(\varphi)) = \bar{r}'(\varphi)] = \begin{cases} p_\epsilon & \text{if } \bar{r}'(\varphi) \in [u, v), \\ p_\epsilon / \exp(\epsilon) & \text{otherwise,} \end{cases}$$

where $p_\epsilon = (\exp(\epsilon) - 1)/\epsilon$ is the sampling probability, and $[u, v)$ is the sampling interval that

$$[u, v) = \begin{cases} \bar{r}(\varphi) + [-C, C) & \text{if } \bar{r}'(\varphi) \in [C, 1 - C), \\ [0, 2C) & \text{if } \bar{r}'(\varphi) \in [0, C), \\ [1 - 2C, 1) & \text{otherwise,} \end{cases}$$

with $C = (\exp(\epsilon)(\epsilon - 1) + 1)/(2(\exp(\epsilon) - 1)^2)$.

Compared with Definition 5 and Definition 6, the redesigned SW mechanisms have a larger p_ϵ and a narrower $[l, r)$ or $[u, v)$ interval for sampling. We can treat the redesigned SW mechanisms as a more aggressive perturbation mechanism: it tries to sample from a narrow dominant sector with a higher probability (p_ϵ), which also leads to a higher probability ($p_\epsilon / \exp(\epsilon)$) of non-dominant sectors due to the LDP constraint.

C.7 Experimental Results for Extensions (Section 3.5.7)

C.7.1 Effect of Different Piecewise-based Mechanisms. Figure 15 compares the performance of different piecewise-based mechanisms for TraCS-D. In addition to the mechanisms defined in Definition 5 and Definition 6, we name the redesigned SW for TraCS-D as TraCS-D (R-SW), as detailed in Appendix C.6. We can observe that TraCS-D consistently exhibits smaller errors than TraCS-D (R-SW) across all the ϵ values. Statistically, the mean AE of TraCS-D is 52.6% of TraCS-D (R-SW) across all the ϵ values. This difference comes from the MSE of their sampling distributions, where Definition 5 and Definition 6 have smaller MSE than R-SW.

C.7.2 TraCS-D for Circular Area. Figure 16 illustrates an example of TraCS-D applied to a circular area $\mathcal{S} = [0, 2\pi) \otimes [0, 1)$. We set the sensitive location τ as $(\pi, 0.5)$ and the privacy parameter $\epsilon = 5$, then collect 50 random samples of perturbed locations using TraCS-D. In this case, the dominant area is $[0.87\pi, 1.13\pi) \otimes [0.32, 0.67)$, as determined by the perturbation mechanisms in Definition 5 and Definition 6. We can observe that the perturbed locations are concentrated in the dominant area.

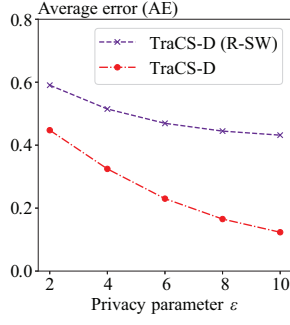


Figure 15: Different piecewise-based mechanisms.

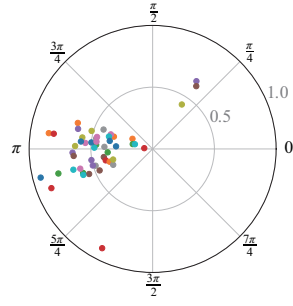


Figure 16: Samples of TraCS-D at $\tau = (\pi, 0.5)$ for circular area.

C.8 2D Laplace Mechanism with Truncation (Section 4.1.1)

For 2D location spaces equipped with the Euclidean distance, a standard Laplace-based approach is the Planar Laplace mechanism [2].[†] It was originally proposed for *geo-indistinguishability*, and it also provides an LDP guarantee with a calibrated sensitivity.

DEFINITION 9 (PLANAR LAPLACE MECHANISM). Let $\mathcal{X} \subseteq \mathbb{R}^2$ be the input domain. The Planar Laplace mechanism $\mathcal{M}_{\text{PL}} : \mathcal{X} \rightarrow \mathbb{R}^2$ is defined by

$$\mathcal{M}_{\text{PL}}(x) = x + \eta,$$

where $\eta \in \mathbb{R}^2$ is a noise vector with density

$$\text{pdf}[\eta] = \frac{\varepsilon^2}{2\pi} \exp(-\varepsilon \|\eta\|_2).$$

Sensitivity calibration for an LDP guarantee. By the triangle inequality, for any $x_1, x_2 \in \mathcal{X}$ and any $y \in \mathbb{R}^2$, we have

$$\frac{\text{pdf}[y - x_1]}{\text{pdf}[y - x_2]} = \exp(\varepsilon(\|y - x_2\|_2 - \|y - x_1\|_2)) \leq \exp(\varepsilon\|x_1 - x_2\|_2).$$

Therefore, over the domain $\mathcal{X}$, the mechanism satisfies $(\varepsilon \cdot \text{diam}(\mathcal{X}))$ -LDP, where $\text{diam}(\mathcal{X})$ is the diameter of $\mathcal{X}$. Equivalently, to ensure ε -LDP for all pairs of inputs in $\mathcal{X}$, one can run the Planar Laplace mechanism with parameter $\varepsilon/\text{diam}(\mathcal{X})$.

Truncation for bounded location spaces. The Planar Laplace mechanism outputs perturbed locations in $\mathbb{R}^2$, which may fall outside the bounded location space $\mathcal{S}$. A common practice is to truncate the output to $\mathcal{S}$, e.g. by projecting it to the nearest point in $\mathcal{S}$. This truncation is a post-processing step and does not affect the LDP guarantee.

C.9 Privacy Parameter Assignment in TraCS-D (Section 4.1.1)

In TraCS-D, the overall perturbation error is jointly determined by the direction perturbation and the distance perturbation. These two components are controlled by ε_d for $\mathcal{M}_o$ and $\varepsilon - \varepsilon_d$ for $\mathcal{M}_\omega$, respectively. As a result, the (theoretical) error bound of TraCS-D depends on both ε and the privacy split ε_d . In principle, the best choice of ε_d is the one that minimizes this bound.

[†]We omit the standard (truncated) Laplace mechanism for $\mathbb{R}$, as it is designed for L_1 distance and has been shown to have worse data utility compared to piecewise-based mechanisms [47].

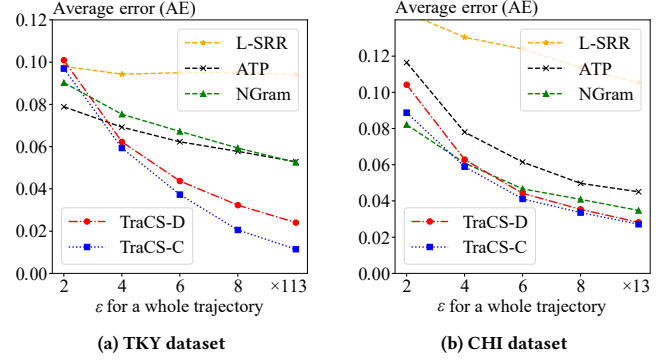


Figure 17: Comparison with trajectory-level ε assignment. The average trajectory length is 113 for TKY and 13 for CHI. TraCS's AE decreases fast as ε increases, outperforming other discrete-space mechanisms when ε is large.

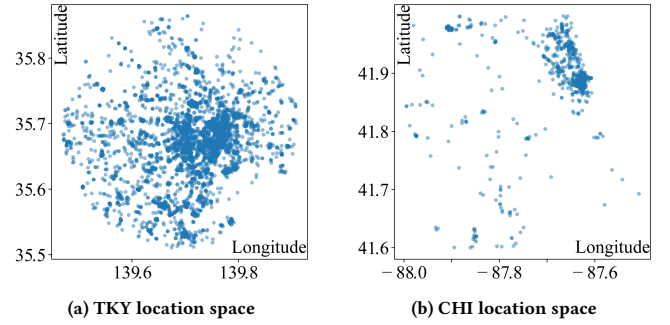
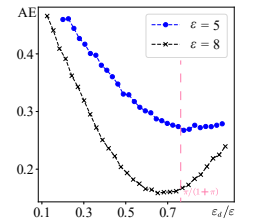


Figure 18: Location spaces for TKY and CHI datasets. TraCS operates over the whole rectangular area that encloses the discrete location space of each city, while other discrete-space mechanisms operate only on the discrete locations (blue points).

However, the optimal split depends on ε and on the shape of $\mathcal{S}$, which makes a universal closed-form choice impractical. Using the same experimental setup as in the Figure 4a, we empirically evaluate how ε_d affects the error of TraCS-D; the results are shown in the figure on the right. We observe a consistent pattern: for each fixed ε , the error decreases initially and then increases as ε_d varies from 0 to the full budget ε .

Across all tested ε values, the minimizer $\varepsilon_d/\varepsilon$ is close to our heuristic split $\varepsilon_d = \pi/(1 + \pi)$ (dashed pink line), which supports the use of this heuristic in practice.



C.10 Privacy Parameter Assignment for a Whole Trajectory (Section 4.2.6)

We compare TraCS with NGram, L-SRR, and ATP under a trajectory-level privacy parameter assignment on the TKY and CHI datasets. The average trajectory lengths are 113 (TKY) and 13 (CHI). Accordingly, we assign a trajectory-level budget of $\varepsilon \times 113$ for TKY

and $\varepsilon \times 13$ for CHI. This scaling makes the trajectory-level setting more comparable to the location-level ε assignment in Figure 9. The results are shown in Figure 17.

We observe that TraCS has larger AEs than the discrete-space mechanisms when ε is small, but its errors decrease rapidly as ε increases. As a result, TraCS outperforms the discrete-space mechanisms in the large- ε regime. Compared with assigning ε at the location level in Figure 9, the overall trends remain similar across all mechanisms: TraCS starts to outperform all other discrete-space mechanisms when $\varepsilon \approx 4$ per location on both datasets. Among the discrete-space mechanisms, ATP achieves the lowest AEs on TKY, whereas NGram (i.e. the Exponential mechanism with a reachable set) attains the lowest AEs on CHI, due to CHI's smaller location space than TKY.

As discussed in Section 4.2.3, the larger average errors (AEs) of TraCS in the small- ε regime can be attributed to its enlarged effective location space. TraCS is designed for continuous spaces and perturbs locations over the rectangular area that encloses the discrete location space of each city, as illustrated in Figure 18. The discrete location set in TKY is comparatively dense and evenly distributed, whereas the location set in CHI is much sparser. Consequently, the gap between the discrete set and its enclosing rectangle is larger for CHI, which amplifies the utility loss of TraCS and explains its weaker performance on CHI (relative to discrete-space mechanisms) compared with TKY.