Summary of Changes

Note: This document uses the quotation environment to indicate changes in the revised `tracs.pdf` file. Within the quotation environment, page numbers and sections are indicated in gray color, changes are highlighted in blue color.

1 Checklist of the Revision Plan

We addressed all the listed items in the revision plan. Specifically, the revision plan associates with sections in this document as follows:

1. We will clarify that we adopt 1D mechanisms from [40] ([20] in this document), but we design new 2D mechanisms tailored for trajectory collection and emphasize privacy, utility, and efficiency aspects of trajectory data. (In the contribution list, Preliminaries, Section 3.2, and Related Work) → Section 2.1
2. We will distinguish between point-level ε and trajectory-level ε, discuss them and clarify we use the former. (In Preliminaries) → Section 2.2
3. We will add an experimental comparison with two-dimensional Laplace mechanisms + truncation. (In Evaluation and Appendix) → Section 2.3
4. We will expand the discussion on the benefits of operating in continuous space, compared with discretization. (In Introduction and Appendix) → Section 2.4
5. We will provide justification for using AE as the primary utility metric and cite additional work in the (L)DP literature to support this. (In Experiments) → Section 2.5
6. We will include trajectory-level privacy comparisons with ATP, NGram, and L-SRR. (In Appendix) → Section 2.6
7. We will clarify how GPS coordinates are treated in this paper and explain the scenarios where Euclidean geometry and spherical geometry are suitable. We will also discuss strategies for handling large-scale spherical spaces. (In Discussions) → Section 2.7
8. We will revise the step calculating the hypotenuse. (In Algorithm 1) → Section 2.8
9. We will discuss how a total privacy budget ε is allocated across the two mechanisms and analyze its impact. (In Appendix) → Section 2.9
10. We will expand the related work, incorporating the papers list in response. (In Related Work) → Section 2.10
11. We will formally define the two evaluation metrics and add experiments under varying hotspot levels. Here we choose a 20% hotspot level (In Experiments) → Section 2.11
12. Other minor comments. We will address additional comments from Reviewers A and D, including notation issues of $M_{1,2}$, and τ, τ' in Algorithm 1, reasons of perturbing direction before distance, etc. → Section 2.12

2 Changes Made in `tracs.pdf`

This section indicates the changes made in the revised `tracs.pdf` file corresponding to the items in the revision plan.

2.1 Novelty and Comparison with OGPM

We clarified that we use the 1D mechanisms from OGPM [20], and discuss the differences in problem setting and contributions between TraCS and OGPM.

> (Contribution list, Page 2) 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 on 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.
>
> (Preliminaries, Page 3) . . . Representative instantiations include OGPM [20] and PM [15] for mean estimation, and SW [10] for distribution estimation. Our perturbation mechanisms build on the 1D piecewise-based mechanism in OGPM to design new 2D mechanisms tailored to trajectory collection.
>
> (Section 3.2, Page 5) We adapt the design of piecewise-based mechanisms for circular domains from OGPM [20]. Specifically, (i) we instantiate a piecewise-based mechanism over the circular domain $[0, 2\pi)$, . . .
>
> (Related Work, Page 16) *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 [20]. Other LDP mechanisms for bounded numerical domains [16, 11, 7] 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 [3, 18, 14], 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.

2.2 Point-Level ε vs Trajectory-Level ε

We clarified that we use point-level ε in this paper, and discussed the difference between point-level and trajectory-level privacy parameters.

> (Preliminaries, Page 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; (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 2.1, 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.9 for completeness.

2.3 Comparison with Two-Dimensional Laplace Mechanisms

We added an experimental comparison with two-dimensional Laplace mechanisms + truncation in Section 4.1 (Continuous space) and detailed the mechanism in appendix.

*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.

> (Evaluations, Page10) 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 the name simplicity, we still refer to this extended method as the strawman method and compare it with TraCS in continuous space. We also include the truncated (planar) Laplace mechanism [1] as a baseline, which perturbs each location by adding noise drawn from a 2D planar Laplace distribution, and then the perturbed location is truncated to the location space $\mathcal{S}$ if it falls outside.[†]

> (Appendix C.7, Page 20) *2D Laplace Mechanism with Truncation.* For 2D location spaces equipped with the Euclidean distance, a standard Laplace-based approach is the Planar Laplace mechanism [1].[‡] It was originally proposed for *geo-indistinguishability*, and it also provides an LDP guarantee with a calibrated sensitivity.

> **Definition 2.1** (Planar Laplace mechanism). *Let $\mathcal{X} \subseteq \mathbb{R}^2$ be the input domain. The Planar Laplace mechanism $\mathcal{M}_{\mathrm{PL}} : \mathcal{X} \to \mathbb{R}^2$ is defined by*

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

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

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

> **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

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

> Therefore, over the domain $\mathcal{X}$, the mechanism satisfies $(\varepsilon \cdot \mathrm{diam}(\mathcal{X}))$-LDP, where $\mathrm{diam}(\mathcal{X})$ 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 $\varepsilon / \mathrm{diam}(\mathcal{X})$.
>
> **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.

> (Section 4.1.2, Page 10) *Error Comparison on Synthetic Datasets.* Figure 1 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.

> (Section 4.1.3, Page 11) *Error Comparison on Real-world Datasets.* Figure 2 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.

2.4 Benefits of Operating in Continuous Space

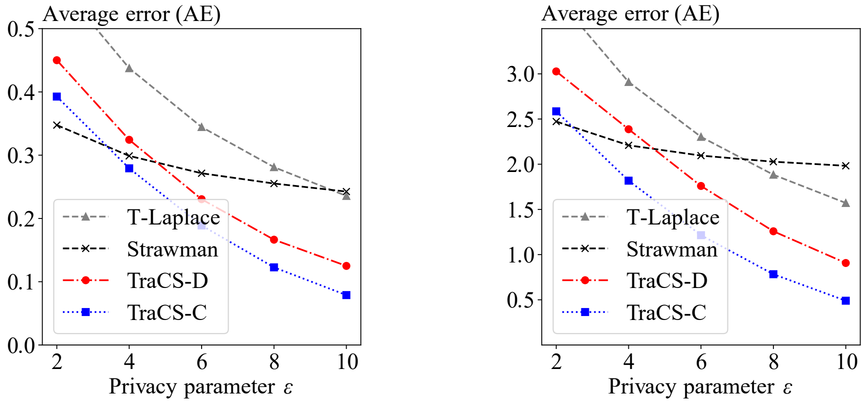
We expanded the discussion on the benefits of operating in continuous space, compared with discretization, in the Introduction and Appendix.

> (Introduction, Page 1)...However, relying on discrete spaces has three issues: (i) Their privacy guarantees are inherently tied to the discrete set. For example, a location space with 10 points implies a weaker *effective* privacy than one with 100 points: even a trivial inference strategy that always outputs a single location succeeds with probability at least 1/10 in the former, regardless of the privacy parameter ε.[§]

[†] Appendix C.7 provides more details on the truncated Laplace mechanism.

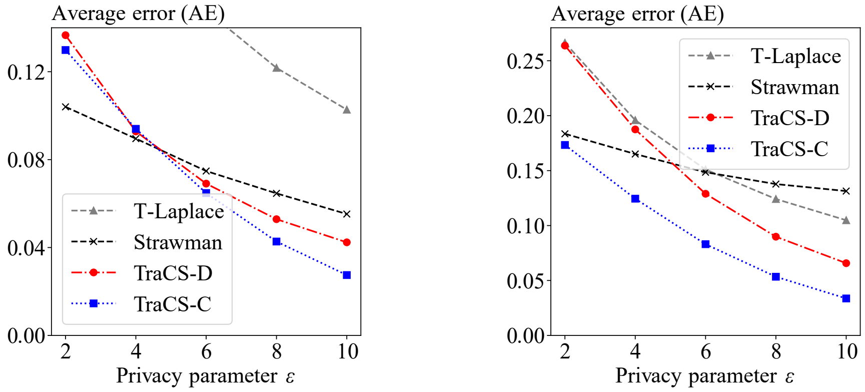
[‡] We omit the standard (truncated) Laplace mechanism for $\mathbb{R}$, as it is designed for L_1 distance and has been shown to have worse data utility compared to piecewise-based mechanisms [20].

[§] Appendix C.1 provides a detailed discussion on the space-dependence of indistinguishability. Appendix C.2 further uses the Exponential mechanism as an example to examine efficiency and data utility limitations of discrete LDP mechanisms.



(a) $\mathcal{S} = [0, 1) \times [0, 1)$ (b) $\mathcal{S} = [0, 2) \times [0, 10)$

Figure 1: Comparison on synthetic datasets.



(a) TKY dataset (b) CHI dataset

Figure 2: Comparison on real-world datasets.

(Appendix C.1, Page 17) *Space-Dependence of Indistinguishability.* 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. $I(x; \mathcal{M}(x)) \leq \mathcal{O}(\varepsilon, |\mathcal{X}|)$, where $|\mathcal{X}|$ denotes the cardinality of the input space $\mathcal{X}$. A similar bound can be found in [5].[¶]

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

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

2.5 Justification for Using AE

We justified the use of AE as the primary utility metric and cited additional related work in the (L)DP literature to support this choice.

> (Evaluation - Setup, Page 10)... where $|\mathcal{T}|$ is the number of locations in the trajectory. A smaller AE indicates better utility of the perturbed trajectory. 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 [12, 19, 18, 3, 14]. Moreover, many other trajectory utility metrics are directly or indirectly related to AE, such as the preservation of range queries and hotspots [3, 18]. For brevity, we defer these additional metrics to Section 4.2.5.

2.6 Comparison with Trajectory-Level ε

We included trajectory-level privacy comparisons with ATP, NGram, and L-SRR in the end of Evaluation and Appendix.

> (Section 4.2.6, Page 13) *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 $\varepsilon/|\mathcal{T}|$, 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.9 provides results and discussion on this setup.

> (Appendix C.9, Page 20) *Privacy Parameter Assignment for a Whole Trajectory.* We compare TraCS with NGram, L-SRR, and ATP under a trajectory-level ε assignment on the TKY and CHI datasets. The average trajectory length is 113 for TKY and 13 for CHI.
>
> The results are shown in Figure 6. We observe that TraCS's absolute errors (AEs) are higher than those of the discrete-space mechanisms when ε is small, but they decrease rapidly as ε increases. Consequently, TraCS outperforms the discrete-space mechanisms when ε is large. As discussed in Section 4.2.3, this is because TraCS is designed for continuous spaces and operates over a rectangular area that encloses a city's discrete location space. This enlarges the effective location space, which can lead to higher errors when ε is small.
>
> Compared with assigning ε at the location level in Figure 9, the overall trends are similar across all mechanisms: TraCS starts to outperform all other discrete-space mechanisms when $\varepsilon \approx 5$ per location, which corresponds to $\varepsilon \approx 2^9$ for a whole trajectory on TKY and $\varepsilon \approx 2^6$

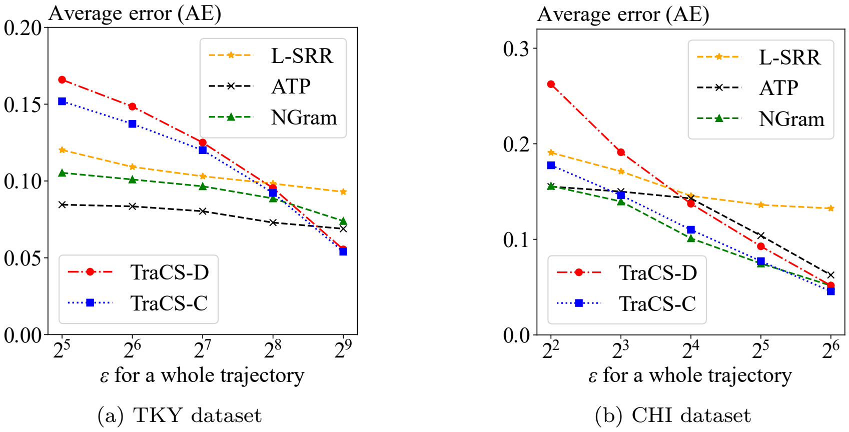


Figure 3: Comparison with trajectory-level ε assignment. The average trajectory length is 113 for TKY and 13 for CHI, TraCS's AE decreases fast as ε increases, outperforming other discrete-space mechanisms when ε is large.

on CHI. Among the discrete-space mechanisms, ATP still achieves the lowest AEs when ε is small on TKY, while NGram (the Exponential mechanism with a reachable set) achieves the lowest AEs when ε is small on CHI. This difference is largely due to the size of the location space: TKY has more locations $(7,798)$ than CHI $(1,000)$, making it harder for the Exponential mechanism to discriminate among locations via their scores.

2.7 Euclidean vs Spherical Geometry

We clarified how GPS coordinates are treated in this paper and explained when Euclidean geometry versus spherical geometry is appropriate. We also discussed practical strategies for handling large-scale spherical domains in the Discussions section.

> (Discussions, Page 8) *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 [13] 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 $(\mathcal{M}_\circ, \mathcal{M}_-)$. 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 $\mathcal{M}_-$, respectively. This design does not involve Euclidean distance and aligns with the semantics of longitude and latitude as separate coordinates.

2.8 Algorithm 1

We revised the step calculating the hypotenuse in Algorithm 1 at Line 7.

```
Algorithm 1: TraCS-D
   Input: Rectangular location space $\mathcal{S}$, sensitive trajectory $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_n\}$, privacy parameter $\varepsilon$
   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     $\varphi \leftarrow \mathrm{atan2}(b_{i+1} - b_i, a_{i+1} - a_i)$;
       ▷ Perturb direction
 5     $\varphi' \leftarrow \mathcal{M}_\circ(\varphi; \varepsilon_d)$;
 6     $R \leftarrow |\mathcal{D}_{r(\varphi)}|$ in Equation (1);
 7     $\overline{r}(\varphi) \leftarrow ||\tau_{i+1} - \tau'_i||_2/R$;                ▷ Sensitive (norm.) distance
 8     $\overline{r}'(\varphi) \leftarrow \mathcal{M}_-(\overline{r}(\varphi); \varepsilon - \varepsilon_d)$;          ▷ Perturb distance
 9     $R' \leftarrow |\mathcal{D}_{r(\varphi')}|$, $r'(\varphi') \leftarrow \overline{r}'(\varphi) \times R'$;                ▷ 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 $\mathcal{T}'$;
```

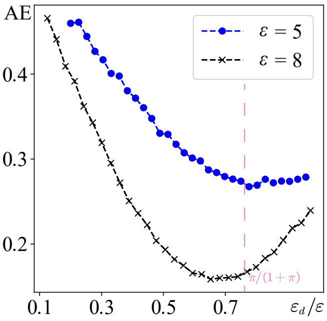
Note that this revision does not affect the use of $\cos(\varphi)$ and $\sin(\varphi)$ in Equation (1), since Equation (1) defines a *set* of feasible values rather than a single deterministic value. In particular, when $\varphi = \pi/2$, using $\cos(\varphi) = 0$ in a denominator would lead to a singularity; in this case, the term involving $\sin(\varphi)$ remains well-defined and yields the correct hypotenuse.

2.9 Privacy Parameter Allocation for ε_d

We discussed how the total privacy budget ε is split between the two sub-mechanisms in TraCS-D, and analyze how this split affects the overall perturbation error in Appendix C.8.

(Appendix C.8, Page 20) *Privacy Parameter Assignment in TraCS-D.*

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 $\mathcal{M}_\circ$ and $\varepsilon - \varepsilon_d$ for $\mathcal{M}_-$, 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.



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 Figure 1a, 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.

2.10 More Related Work and Discussions

We incorporated several recently published papers on LDP and trajectory collection into the Related Work section.

(Related Work, Page 16) 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. WF-LDPSR [9] 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 [17]. In addition, recent work studies poisoning attacks against NGram and ATP [8], where adversaries induce malicious users to submit poisoned trajectories to the data collector with the goal of promoting target patterns.

(Related Work, Page 16) Another line of research is trajectory *synthesis* for privacy-preserving trajectory publication. A typical work is LDPTrace [4], . . . Following LDPTrace, ADGTrace [2] 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.

2.11 Trajectory Utility Metrics and Hotspot Levels

We formally defined the two evaluation metrics, and following the suggested hotspot setting, use the hotspot level in our experiments on the top 20% most frequent locations (instead of the top 50% in the original submission).

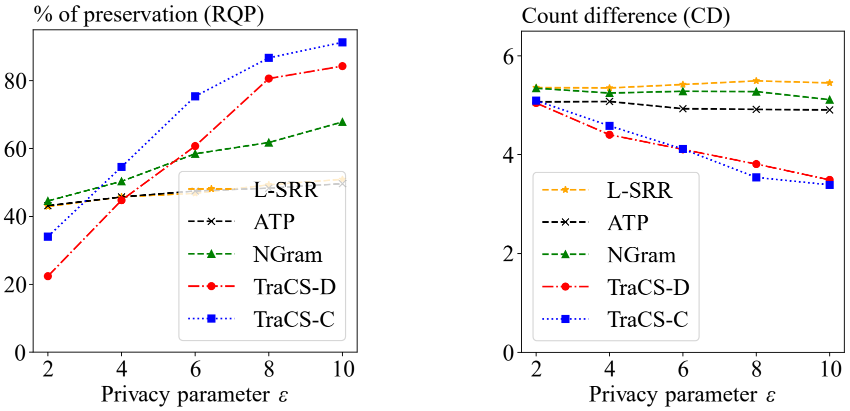
(Section 4.2.5, Page 13) *Experimental Results for Other Metrics.* This subsection evaluates the performance of TraCS-D and existing methods under two additional metrics: range query preservation (Formula 17 in [3]) and hotspot preservation (Section 6.2.4 in [18]). Given a threshold δ, a perturbed location 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:

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

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:

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

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.



(a) Range query preservation

(b) Hotspot preservation

Figure 4: Comparison of other trajectory metrics on the CHI dataset. Higher range query preservation and lower error in hotspot preservation indicate better performance.

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 4. It can be seen that TraCS generally outperforms existing methods across different privacy levels, demonstrating superior 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.

2.12 Other Comments

The notion of $\mathcal{M}_{1,2}$ is clarified in the Preliminaries section:

Perliminaries, Page 2

Theorem 2.1 (Sequential Composition of LDP [6, 18]). *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two mechanisms that satisfy ε_1 and ε_2-LDP, respectively. Their composition, defined as $\mathcal{M}_{1,2} := (\mathcal{M}_1, \mathcal{M}_2) : (\mathcal{X}_1, \mathcal{X}_2) \rightarrow (Range(\mathcal{M}_1), Range(\mathcal{M}_2))$, satisfies $(\varepsilon_1 + \varepsilon_2)$-LDP.*

Algorithm 2: TraCS-D

```
Input: Rectangular location space $\mathcal{S}$, sensitive trajectory $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_n\}$, privacy parameter $\varepsilon$
Output: Perturbed trajectory $\mathcal{T}' = \{\tau'_1, \tau'_2, \ldots, \tau'_n\}$
$\mathcal{T}' \leftarrow \emptyset, \mathcal{T} \leftarrow \tau'_0 \cup \mathcal{T}$;                ▷ Add a dummy location $\tau'_0$
for $i \leftarrow 0$ to $n-1$ do
    ▷ $\tau'_i$ is the ref location for this iteration
    $\tau'_i = (a_i, b_i), \tau_{i+1} = (a_{i+1}, b_{i+1})$;
    ▷ Sensitive direction
    $\varphi \leftarrow \mathrm{atan2}(b_{i+1} - b_i, a_{i+1} - a_i)$;
    ▷ Perturb direction
    $\varphi' \leftarrow \mathcal{M}_\circ(\varphi; \varepsilon_d)$;
    $R \leftarrow |\mathcal{D}_{r(\varphi)}|$ in Equation (1);
    $\overline{r}(\varphi) \leftarrow ||\tau_{i+1} - \tau'_i||_2 / R$;                ▷ Sensitive (norm.) distance
    $\overline{r}'(\varphi) \leftarrow \mathcal{M}_-(\overline{r}(\varphi); \varepsilon - \varepsilon_d)$;                ▷ Perturb distance
    $R' \leftarrow |\mathcal{D}_{r(\varphi')}|$, $r'(\varphi') \leftarrow \overline{r}'(\varphi) \times R'$;                ▷ De-normalize
    ▷ Transform back to (longitude, latitude)
    $\tau'_{i+1} = (a_i + r'(\varphi')\cos(\varphi'), b_i + r'(\varphi')\sin(\varphi'))$;
    $\mathcal{T}' \leftarrow \mathcal{T}' \cup \tau'_{i+1}$;                ▷ $\tau'_{i+1}$ is the next ref location
return $\mathcal{T}'$;
```

In Algorithm 1, the revised version consistently uses τ'_i as the reference location when perturbing τ_{i+1} in each iteration. (Shown above)

We add a discussion explaining why TraCS-D perturbs the direction before the distance in the Discussions section.

> (Discussions, Page 9) *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, more direct mechanism design.

3 The Second Revision

3.1 Benefits of Operating in Continuous Space compared with Discretization

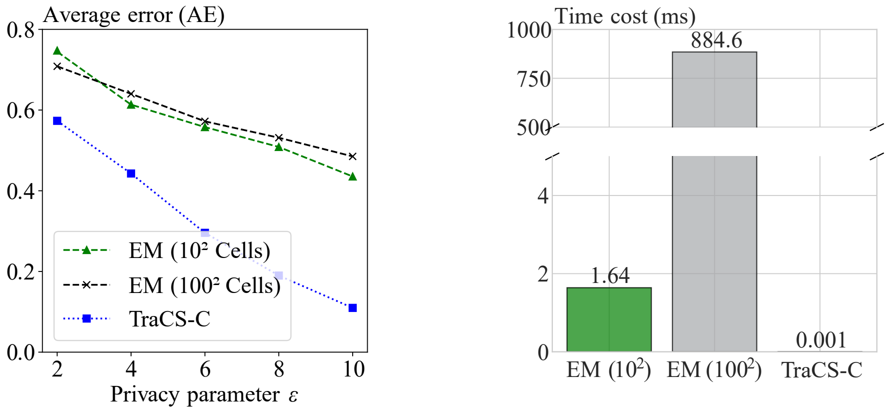
We expanded the discussion of the benefits of operating in continuous space in terms of efficacy, efficiency, and applicability relative to discretization, in the Introduction and the Appendix. Specifically, (i) we reframed the discussion of the Exponential Mechanism (previously in Section 3.2) to directly support the efficacy and efficiency claims in the Introduction. (ii) We also added Appendix C.3, which provides a more detailed discussion of the challenges of discretizing continuous spaces via gridding, including the difficulty of balancing privacy, utility, and computational cost.

> (Introduction, Page 1)...Moreover, the widely used Exponential mechanism has linear sampling complexity in the domain size, making each perturbed location expensive to generate.[‖] (iii) Discrete methods are not directly applicable to inherently continuous location spaces, such as flying and sailing trajectories or sensor trajectories from wearable devices. Although one can discretize a continuous space and then apply discrete methods, doing so inherits the aforementioned issues. Moreover, selecting a suitable discretization strategy is non-trivial to balance privacy, utility, and computational cost.[**]

> (Appendix C.3, Page 18) *Limitations of Applying Discrete LDP Mechanisms to Continuous Spaces.* Although a continuous space can be discretized before applying discrete LDP

[‖] 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.



(a) Error of the gridding approach with different cell sizes.

(b) Average time cost of perturbing one location.

Figure 5: 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.

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 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 average error and the average time required to perturb a single location. Figure 5 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.

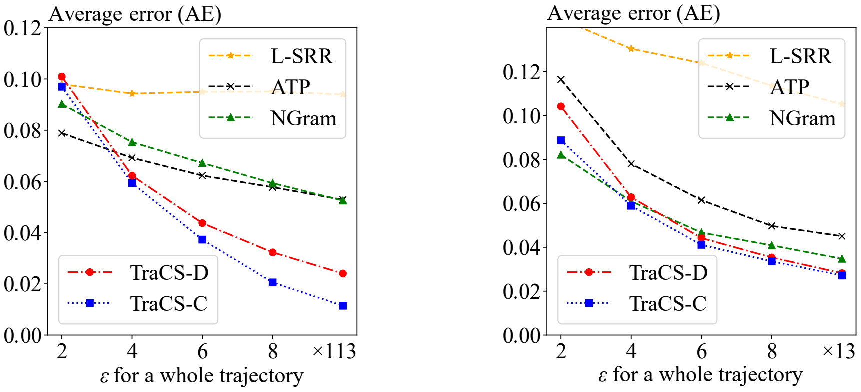
3.2 Comparison with Trajectory-Level Epsilon

In the previous version, we used a power-of-two scale on the ε x-axis. Because this scale does not correspond linearly to the per-location ε used in Figure 9, it may cause confusion when comparing these results with Figure 9 (location-level ε). To improve visual consistency, we revised the x-axis to a linear scale by multiplying ε by the average trajectory length, and we added a more detailed discussion of the performance trends.

(Appendix C.10, Page 21) *Privacy Parameter Assignment for a Whole Trajectory.* 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 8. The results are shown in Figure 6.

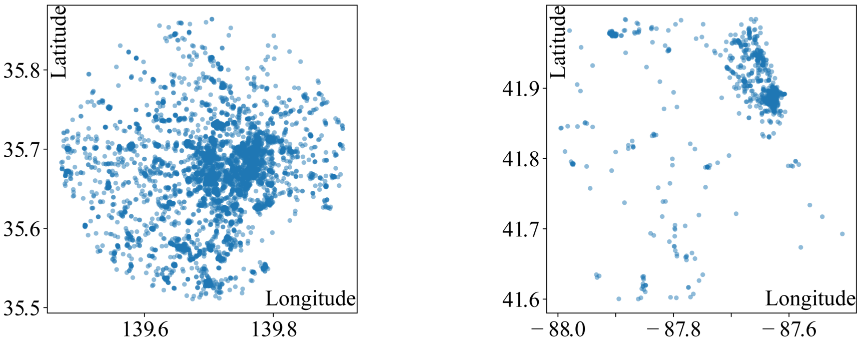
We observe that TraCS has larger AEs than the discrete-space mechanisms when ε is small, but its errors decrease rapidly as ε increases. As a result, TraCS outperforms the discrete-space mechanisms in the large-ε regime. Compared with assigning ε at the location level in Figure 8, the overall trends 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.



(a) TKY dataset

(b) CHI dataset

Figure 6: 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.



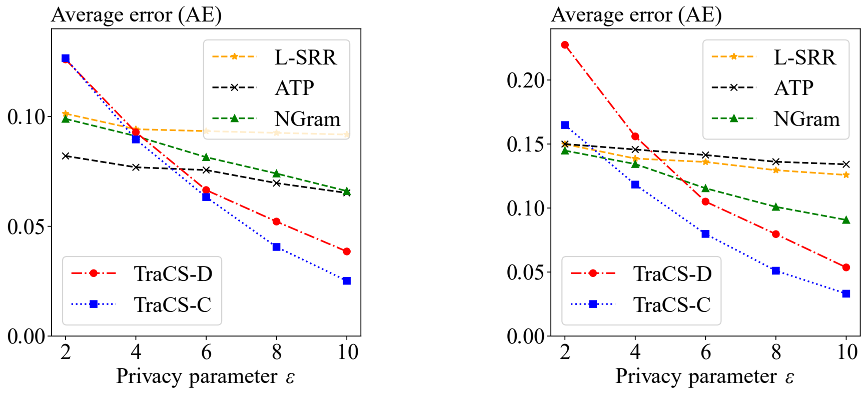
(a) TKY location space

(b) CHI location space

Figure 7: 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).

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 7. 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.

Figure 9 (Figure 8 in this document) is attached below for easier visual comparison with the new Figure 6. Note that the trends may differ because trajectories have varying lengths. In particular, the CHI dataset exhibits higher length variance, which can induce different effective ε values across trajectories under the trajectory-level assignment.



(a) TKY dataset

(b) CHI dataset

Figure 8: (Figure 9 in tracs.pdf) Comparison on real-world datasets. TraCS operates on rectangular areas encompassing the city, which may result in larger AE when ε is small.

References

[1] Miguel E. Andrés, Nicolás Emilio Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: differential privacy for location-based systems. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 901–914. ACM, 2013.

[2] Hui Cai, Chen Lan, Biyun Sheng, Jian Zhou, Yuanyuan Yang, Yanmin Zhu, and Fu Xiao. Adgtrace: Achieving adaptive trajectory synthesis with generated data. *IEEE Trans. Mob. Comput.*, 25(1):148–163, 2026.

[3] Teddy Cunningham, Graham Cormode, Hakan Ferhatosmanoglu, and Divesh Srivastava. Real-world trajectory sharing with local differential privacy. *Proc. VLDB Endow.*, 14(11):2283–2295, 2021.

[4] Yuntao Du, Yujia Hu, Zhikun Zhang, Ziquan Fang, Lu Chen, Baihua Zheng, and Yunjun Gao. Ldptrace: Locally differentially private trajectory synthesis. *Proc. VLDB Endow.*, 16(8):1897–1909, 2023.

[5] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy, data processing inequalities, and statistical minimax rates. *arXiv: Statistics Theory*, 2013.

[6] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[7] Naoise Holohan, Spiros Antonatos, Stefano Braghin, and Pól Mac Aonghusa. The bounded laplace mechanism in differential privacy. *J. Priv. Confidentiality*, 10(1), 2020.

[8] I-Jung Hsu, Chih-Hsun Lin, Chia-Mu Yu, Sy-Yen Kuo, and Chun-Ying Huang. Poisoning attacks to local differential privacy protocols for trajectory data. *CoRR*, abs/2503.07483, 2025.

[9] Yan-zi Li, Li Xu, Jing Zhang, and Liao-ru-xing Zhang. WF-LDPSR: A local differential privacy mechanism based on water-filling for secure release of trajectory statistics data. *Comput. Secur.*, 148:104165, 2025.

[10] Zitao Li, Tianhao Wang, Milan Lopuhaä-Zwakenberg, Ninghui Li, and Boris Skoric. Estimating numerical distributions under local differential privacy. In David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo, editors, *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, pages 621–635. ACM, 2020.

[11] Fang Liu. Statistical properties of sanitized results from differentially private laplace mechanism with univariate bounding constraints. *Trans. Data Priv.*, 12(3):169–195, 2019.

[12] Àlex Miranda-Pascual, Patricia Guerra-Balboa, Javier Parra-Arnau, Jordi Forné, and Thorsten Strufe. Sok: Differentially private publication of trajectory data. *Proc. Priv. Enhancing Technol.*, 2023(2):496–516, 2023.

[13] Montana State University. Lat/Lon and UTM Conversion - Yellowstone Research Coordination Network, n.d.

[14] Han Wang, Hanbin Hong, Li Xiong, Zhan Qin, and Yuan Hong. L-SRR: local differential privacy for location-based services with staircase randomized response. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 2809–2823. ACM, 2022.

[15] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. 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*, pages 638–649. IEEE, 2019.

[16] Benjamin Weggenmann and Florian Kerschbaum. Differential privacy for directional data. In Yongdae Kim, Jong Kim, Giovanni Vigna, and Elaine Shi, editors, *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, pages 1205–1222. ACM, 2021.

[17] Dongyue Zhang, Weiwei Ni, Nan Fu, Lihe Hou, and Ruyu Zhang. Locally differentially private trajectory publication based on regional popularity awareness. *IEEE Trans. Inf. Forensics Secur.*, 20:6719–6732, 2025.

[18] Yuemin Zhang, Qingqing Ye, Rui Chen, Haibo Hu, and Qilong Han. Trajectory data collection with local differential privacy. *Proc. VLDB Endow.*, 16(10):2591–2604, 2023.

[19] Xiaodong Zhao, Dechang Pi, and Junfu Chen. Novel trajectory privacy-preserving method based on clustering using differential privacy. *Expert Syst. Appl.*, 149:113241, 2020.

[20] Ye Zheng, Sumita Mishra, and Yidan Hu. Optimal piecewise-based mechanism for collecting bounded numerical data under local differential privacy. *Proc. Priv. Enhancing Technol.*, 2025(4):146–165, 2025.