applied sciences

MDPI

Article

Releasing Differentially Private Trajectories with Optimized Data Utility

Bing Li, Hong Zhu * and Meiyi Xie

School of Computer Sciences and Technology, Huazhong University of Science and Technology, Wuhan 430074, China; bing@hust.edu.cn (B.L.); xiemeiyi@hust.edu.cn (M.X.)
* Correspondence: zhuhong@hust.edu.cn

Abstract: The ubiquity of GPS-enabled devices has resulted in an abundance of data about individual trajectories. Releasing trajectories enables a range of data analysis tasks, such as urban planning, but it also poses a risk in compromising individual location privacy. To tackle this issue, a number of location privacy protection algorithms are proposed. However, existing works are primarily concerned with maintaining the trajectory data geographic utility and neglect the semantic utility. Thus, many data analysis tasks relying on utility, e.g., semantic annotation, suffer from poor performance. Furthermore, the released trajectories are vulnerable to location inference attacks and de-anonymization attacks due to insufficient privacy guarantee. In this paper, to design a location privacy protection algorithm for releasing an offline trajectory dataset to potentially untrusted analyzers, we propose a utility-optimized and differentially private trajectory synthesizer (UDPT) with two novel features. First, UDPT simultaneously preserves both geographical utility and semantic utility by solving a data utility optimization problem with a genetic algorithm. Second, UDPT provides a formal and provable guarantee against privacy attacks by synthesizing obfuscated trajectories in a differentially private manner. Extensive experimental evaluations on real-world datasets demonstrate UDPT's outperformance against state-of-the-art works in terms of data utility and privacy.

Keywords: location-based services; mobile computing; data release; location privacy protection; data utility

check for updates

Citation: Li, B.; Zhu, H.; Xie, M. Releasing Differentially Private Trajectories with Optimized Data Utility. *Appl. Sci.* **2022**, *12*, 2406. https://doi.org/10.3390/app12052406

Academic Editor: Amalia Miliou

Received: 26 January 2022
Accepted: 21 February 2022
Published: 25 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The ubiquity of GPS-equipped devices, from airplanes and automobiles to smartphones and wearable technology, along with the popularity of location-based services (LBSs), such as automobile navigation and searching nearby restaurants, has greatly eased the collection of individual trajectories, where a trajectory is a sequence of locations visited by an individual over time. To help researchers develop solutions for a wide range of important problems, such as business decision-making and urban planning [1], data curators often publish individual trajectories to third-party data analyzers , which is known as offline release of individual trajectories. For instance, New York City Taxi and Limousine Commission publicly releases a trajectory dataset of taxi passengers every month. The data analyzers, such as urban planners, can improve the community division with the help of the spatial–temporal regularity of human movement patterns [2]. However, such data release poses a serious threat to individual location privacy, since (potentially) untrusted analyzers may have great interest in deriving personal identity and sensitive locations from the individual trajectories [3].

To mitigate the threat, actual trajectories are usually obfuscated by location privacy protection methods (LPPMs), e.g., location perturbation [4], cryptography [5], trajectory synthesis [6], before being released. Among these works, the trajectory synthesis has been widely accepted for offline releasing trajectories because of its good preservation of population mobility patterns [6] . Generally, it builds a trajectory generator fitting the movement patterns of the actual (or original) trajectories and then synthesizes a dataset

of obfuscated trajectories in a generative fashion. To protect location privacy, the process of trajectory synthesis is perturbed by random noise calibrated by some privacy notions, e.g., differential privacy [7], a de facto notion for protecting privacy due to its strong privacy guarantee. However, existing works based on trajectory synthesis can neither well preserve the data utility nor effectively thwart the privacy attacks. The reasons are as follows.

First, existing location privacy protection algorithms merely focus on the geographical utility in trajectories while neglecting the simultaneous preservation of semantic utility, resulting in poor performance of numerous data analysis tasks. Specifically, the data utility in individual trajectories consists of two aspects, namely geography and semantics. The geographical utility represents the superficially spatial–temporal regularity of population movement, e.g., trajectory diameter distribution [6]. Early trajectory data analysis tasks solely rely on the geographical utility. In this case, existing location privacy protection algorithms can already satisfy the analyzer's requirement. However, recently, a large number of popular analysis tasks have emerged, e.g., semantic annotation [8], trajectory prediction [9] , providing great convenience to people's daily life. These emerging data analysis tasks not only require the geographical utility but also heavily rely on deep-level regularity implied in the individual trajectories, especially the semantics that motivates individual movement, e.g., periodic movement patterns [9] and location categories [10]. Nonetheless, simultaneously preserving both kinds of data utility on the condition of ensuring location privacy is an intractable problem since there exist conflicts between them [10]. For instance, AdaTrace [6] protects location privacy by differentially private trajectory synthesis, which solely focuses on the geographical utility while neglects the location semantics that motivates human movement. As a result, the data analysis tasks such as semantic annotation suffer from low precision.

Second, existing location privacy protection algorithms cannot provide a strong privacy guarantee due to the negligence of privacy attacks using data utility as side channels. For instance, the geographical utility could reveal the individual identity, so individuals could be distinguished from each other, which is known as de-anonymization attacks, e.g., four spatial–temporal points are enough to uniquely identify 95% of the anonymized individuals [11]. Most of existing work cannot thwart the de-anonymization attacks because they protect the single location without a comprehensive consideration of whole the trajectory. For example, MLCE [12] protects location privacy by independently perturbing each actual location into a nearby location. In addition, the semantic utility could reveal individuals' sensitive locations such as home and workplace, resulting in location inference attacks [3]. Figure 1 depicts an example where the periodic movement patterns and location categories, two types of semantic utility, are used to deduce home and workplace. Existing work cannot provide a formal and provable privacy guarantee to thwart the privacy attacks. For instance, Tian et al. [13] protects privacy by blurring the actual location into a k-anonymous area. Since there is not provable correlation between the privacy parameter k and the privacy guarantee, the data curator cannot determine k for a specific scenario.

In this work, given an actual trajectory dataset, we aim to address the following problem, i.e., designing a location privacy protection algorithm for releasing an offline and obfuscated version on the conditions that (a) simultaneous preservation of both geographical as well as semantic utility of the actual data, and (b) effective prevention from the location inference and de-anonymization attacks. To tackle this issue, we propose a utility-optimized and differentially private trajectory synthesis algorithm named UDPT, which is composed of three sequential phases as follows. (i) To provide an effective defense against the location inference attacks, locations in the actual trajectories are blurred into regions by private location clustering. (ii) To simultaneously preserve both the geographical utility and semantic utility, we privately select a sequence of utility-optimized candidate obfuscated location sets from the clusters. We model the selection as a multi-objective optimization problem and tackle it with a differentially private genetic algorithm. (iii) Based on the utility-optimized candidate obfuscated location sets, to defend against the de-anonymization attacks, we construct an obfuscated trajectory synthesizer based on

the Conditional Random Fields (CRF) and privately select the (final) obfuscated trajectories by the sequence decoding of CRF. We conclude the contributions as follows:

- We propose a location privacy protection algorithm, UDPT. It enables the data curator to release an offline trajectory dataset for data-mining purposes in a utility-optimized and differentially private manner. The data analysis applications, e.g., semantic annotation and trajectory prediction, which heavily rely on trajectory data geographical utility as well as the semantic utility can be benefited.
- UDPT can preserve the data utility in terms of geography as well as semantics, simultaneously, by a multiple-objective optimization algorithm, so the released dataset can improve the performance of numerous data analysis tasks.
- UDPT can provide a formal and provable privacy guarantee by differentially private and generative trajectory synthesis. To our best knowledge, it is the first work which ensures differential privacy and preserves both types of data utility.
- Extensive evaluations on real-world datasets demonstrate that UDPT not only can outperform state-of-the-art works, in terms of multiple data utility metrics, but can also effectively prevent the de-anonymization attacks and location inference attacks.

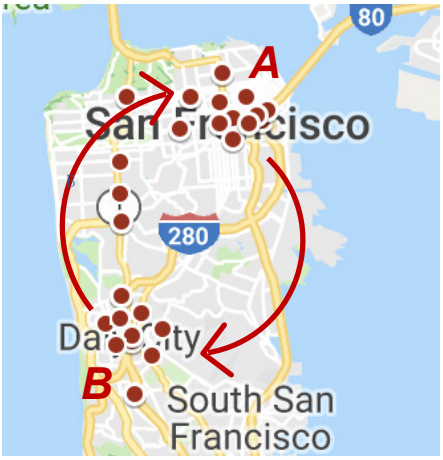

Figure 1. An illustrative example of the privacy threat caused by location semantics, where red points represent the locations that an individual visited. Especially, two regions where points concentrate are represented by red letters *A* and *B* respectively. We observe two periodic movement patterns, i.e., leaving from *A* to *B* and returning periodical. Suppose that the categories of *A* and *B* are residential district and business district, respectively. Then, *A* and *B* have a great chance to be home and workplace, respectively.

The structure of this work is as follows. We summarize related work in Section 2. Section 3 presents preliminaries, including problem statement and the differential privacy. In Section 4, we elaborate on three main phases of UDPT. Experimental evaluations on privacy and data utility are presented in Section 5. We conclude and discuss future work in Section 6. Details of the implementation of UDPT are shown in Appendix A.

2. Related Work

We summarize the existing works related to location privacy from two aspects, namely location privacy attacks and location privacy protection algorithms, where the latter are further categorized into three classes below. We discuss relevant work under each category. Table 1 presents a summary of the related location privacy protection algorithms.

2.1. Location Privacy Attacks

A location inference attack refers to an adversarial action where an attacker, e.g., a curious data analyzer, tries to infer actual locations over the locations perturbed by location privacy protection algorithms. For instance, Shokri et al. use a hidden Markov model to capture individual mobility patterns in the obfuscated trajectory and infer the actual

trajectory by solving the decoding problem of the model. Li et al. [3] argue that actual locations could be revealed with spatial–temporal-social–semantic correlations as side channels. For example, speed limit, a type of temporal correlation, could be used to exclude the unreachable (dummy) locations at the next time instant if the adversary knew the current actual location.

A de-anonymization attack, a.k.a., re-identification, refers to an adversarial action where an attacker intends to deduce individual identity over anonymized trajectories. Generally, the attacker is assumed to have access to the public trajectories of some target individuals of whom the identities are known, together with a number of anonymized trajectories containing private information such as sensitive locations. The aim of the attack is to find a true match between the background trajectories and the anonymized ones using individual mobility patterns, then the private information of the target individuals would be revealed. For example, Montjoye et al. [11] find that 4 spatial–temporal points are enough to uniquely identify 95% of the anonymized individuals. In the subsequent work [14], Naini et al. propose a de-anonymization attack based on the visit frequency of locations. Recently, Drakonakis et al. [15] show that Twitter individuals that do not provide their full name could still be de-anonymized and their sensitive locations, such as home and workplace could be deduced.

Table 1. Summary of related work in terms of location privacy protection.

Work	Method	Privacy Notion	Privacy Parameters	Preserved Utility	Thwarted Attacks
Oya 2017 [12]	location perturbation	differential privacy	ϵ	geographical utility	location inference
Tian 2021 [13]	location perturbation	k-anonymity	k	geographical + semantic utility	location inference
Xu 2021 [16]	location perturbation	dummy location + cache	\	geographical utility	location inference
Huang 2022 [4]	location perturbation	k-anonymity	k	geographical utility	location inference
Schlegel 2017 [17]	cryptography	order-retrievable encryption	\	geographical utility	location inference
Guan 2021 [18]	cryptography	oblivious transfer	\	geographical utility	location inference
Qureshi 2022 [5]	cryptography	blockchain	\	geographical utility	location inference
He 2015 [19]	trajectory synthesis	differential privacy	ϵ	geographical utility	location inference
Bindschaedler 2016 [20]	trajectory synthesis	plausible deniability	k, δ	geographical + semantic utility	location inference + de-anonymization
Gursoy 2018 [6]	trajectory synthesis	differential privacy	ϵ	geographical utility	location inference + de-anonymization
Tan 2020 [21]	trajectory synthesis	k-anonymity	k	semantic utility	location inference
Sina 2021 [22]	trajectory synthesis	k-anonymity	k	geographical utility	location inference
This work	trajectory synthesis	differential privacy	ϵ	geographical + semantic utility	location inference + de-anonymization

2.2. Location Perturbation

Location perturbation usually indicates perturbing an actual location by blurring it into an area, replacing it with another location, or combining it with some indistinguishable dummy locations. Location privacy protection algorithms based on location perturba-

tion can be applied to the scenarios where individual locations are continuously released, e.g., online location-based services. However, locations in the trajectory are perturbed independently and the spatial–temporal correlations within the whole trajectory are neglected, so privacy could be compromised by adversaries who have access to the correlations [23].

An implementation of location perturbation is to blur the actual location into an area. For example, Tian et al. [13] investigate the problem of privately releasing location data in an online manner. Given an actual location at some time instant, the authors blur the location into an area containing other $k-1$ locations, which have similar geographical and semantic utility with the actual location. To improve the data utility, they model the trade-off between privacy and data utility as an optimization problem and solve it by an improved multi-objective optimization algorithm. However, the authors focus on the data utility of a single location while neglecting that of whole the trajectory. In addition, independently protecting each actual location without a comprehensive consideration of the correlations in the trajectory was subject to location inference attacks and de-anonymization attacks [23]. Huang et al. [4] discover a specific applying scenario of location privacy protection, ride-sharing. They propose a private-protecting location release scheme called pShare, which applies a zone-based travel time estimation approach to reduce ride-sharing detouring waste while hiding each rider's actual location in the zone. However, changing the format of location data from point to area could lead to modification of the data analysis interfaces.

Another implementation of location perturbation is to replace the actual location with another location. For instance, Oya et al. [12] integrate the differential privacy with a privacy metric called conditional entropy to decrease adversary's correctness, together with a utility metric named worst loss for ensuring data utility. A remapping based on both metrics is appended to the output of the differential privacy, so obfuscated locations with higher utility, as well as higher privacy levels, can be produced.

Instead of modifying the actual locations, another implementation of location perturbation is to combine the actual location with some indistinguishable dummy locations. For instance, to protect the actual locations of drivers on the internet of vehicles from being revealed by road restriction, Xu et al. [16] propose a dummy-generation based location perturbation methods. The perturbed location sent to the data analyzer is composed of the actual location and several nearby fake locations following the road restriction, so it is hard for the adversary to distinguish the actual location from the dummies.

2.3. Cryptography

The location privacy protection algorithms based on cryptography mainly exploit encryption techniques to hide actual trajectories from untrusted analyzers. They are usually used in scenarios such as online data releasing. The advantage is providing a provable privacy guarantee. However, people usually need to develop ad hoc encryption protocols for specific application scenarios. In addition, cryptography techniques usually bring about high computation costs and network overhead.

For example, Guan et al. [18] propose an oblivious-transferring and k-nearest neighbor query scheme based on the modified Paillier cryptosystem, in which the LBS server cannot link two queries even if they are initiated by individuals at the same location. However, the encryption protocols take a significantly long time to initialize queries and responses, which makes them impractical to be deployed in mobile devices. Schlegel et al. [17] propose a new encryption notion, order-retrievable encryption, to enable individuals to share their actual locations without leaking any private information to the LBS server. Recently, Qureshi et al. [5] propose a blockchain-based and privacy-preserving mechanism for the Internet of vehicle networks, which allows drivers to hide their exact locations and takes control of their data during the data communication and voting process.

Although the cryptography technique can be used to protect location privacy, it cannot be directly compared with the location perturbation and trajectory synthesis techniques because the privacy notions, attacker assumptions and models are different.

2.4. Trajectory Synthesis

Trajectory synthesis is essentially an obfuscated trajectory generator, which fits the mobility patterns of actual trajectories and generates obfuscated trajectories in a private-protecting way. The location privacy protection algorithms based on trajectory synthesis are often used in scenarios such as offline data releasing. To protect privacy, the synthesis process will be perturbed under the guidance of some privacy notions such as k-anonymity or differential privacy.

For example, He et al. [19] first propose an end-to-end trajectory data releasing solution named DPT. It discretizes the area in which individuals move by a uniform grid and aggregates actual trajectories to grid trajectories. Based on that, a trajectory generator using a prefix tree is constructed and then a number of trajectories are generated. The advantage of DPT is good preservation of geographical utility, especially the moving speed and directions, due to the exploitation of multiple-level granularity grid (a.k.a. hierarchical reference systems). The trajectory generator is perturbed by injecting random noise into the prefix tree to ensure differential privacy. In subsequent work [6], Gursoy et al. argue that solely relying on differential privacy cannot provide enough guarantee to thwart privacy attacks such as de-anonymization, since the obfuscated trajectories generated by previous works are still distinguishable. Therefore, they propose an attack-resilient trajectory releasing method called AdaTrace. First, in contrast with the DPT's uniform grid, AdaTrace discretizes the moving area by a location-density-aware grid to capture the complex spatial density and locality distributions, two types of the geographical utility. Second, the authors exploit a Markov model to capture the location transition patterns in actual trajectories. Third, a number of obfuscated trajectories are point-wisely sampled from the grid trajectories, where the sampling is perturbed by random noise calibrated by differential privacy. Finally, AdaTrace checks whether the indistinguishability of the obfuscated trajectories satisfies the given requirement.

Except for the differential privacy, k-anonymity is also a widely used privacy notion to thwart privacy attacks. For instance, Tan et al. [21] find that individual location privacy could be compromised when the sensitive semantics of locations, e.g, hospitals, are derived. To address this problem, the authors propose a k-anonymous privacy protection algorithm for releasing semantic trajectory data, which blurs an actual location, together with other $k-1$ locations sharing similar semantics, into an anonymous area. The advantage is a well preservation of semantic utility. However, it independently anonymizes each actual location while taking into no consideration the data utility of whole the actual trajectory. Sina et al. [22] propose to apply machine learning algorithms for clustering the actual trajectories and randomly sample the obfuscated trajectories to ensure that every trajectory in the obfuscated trajectory dataset is indistinguishable from at least $k-1$ other trajectories.

Bindschaedler et al. [20] propose another privacy notion, statistical dissimilarity, measuring the indistinguishability between trajectories, which can simultaneously preserve data utility in terms of both geography and semantics. However, compared with differential privacy, the proposed privacy notion cannot provide a formal and provable guarantee of location privacy, so it will be hard for the data curators to determine a suitable privacy level for a certain data releasing task. In contrast, the privacy level, ϵ, of the differential privacy usually takes values in a widely-accepted interval, i.e., $\epsilon \in [0.1, 10]$, where less ϵ indicates a higher privacy level.

Compared with the location perturbation and cryptography, trajectory synthesis enables better preservation of data utility, especially the population mobility patterns, since it takes into consideration the global features and spatial–temporal–semantic correlations among locations in the actual trajectories. However, existing location privacy protection algorithms based on trajectory synthesis consider only the geographical utility rather than the semantic utility, resulting in low performance of emerging data analysis tasks such as semantic annotation [8], trajectory prediction [9].

3. Preliminaries

In this section, we present notations used in this paper, together with an introduction to differential privacy. Table 2 shows the notations that are frequently used in this paper.

Table 2. List of notations.

Symbol	Meaning
u	individual
$\mathcal{U}$	set of individuals
$\mathcal{T}$	set of time instants
$\mathcal{L}$	geographical space
$\boldsymbol{l}$	trajectory
l_t	location record at the time instant t
D_A	actual trajectory dataset
D_O	obfuscated trajectory dataset
f	location privacy protection algorithm
$P(f)$	privacy metric
$Q(f)$	utility metric

3.1. Problem Statement

Notations. Consider that a set of individuals $\mathcal{U} = \{u_1, u_2, \dots\}$ move in a geographic space, which is denoted by a set of discrete geographical locations, $\mathcal{L} = \{l_1, l_2, \dots\}$, where a location is represented by a vector of GPS coordinates , e.g., $(-73.989308, 40.741895)$. The (actual) trajectory of any individual $u \in \mathcal{U}$ is represented by a sequence of location records, $\boldsymbol{l} = (l_1, l_2, \dots, l_t, \dots)$, which u has visited over a period of time $\mathcal{T} = \{1, 2, 3, \dots\}$, where l_t records u's location at the time instant $t \in \mathcal{T}$, $l_1, l_2, \dots \in \mathcal{L}$.

Problem to tackle. Suppose that a data curator , e.g., the New York City Taxi & Limousine Commission [2], has collected a dataset of actual trajectories of the individuals, $D_A = \{\boldsymbol{l}_u | u \in \mathcal{U}\}$. The data curator would like to publish the dataset to some (potentially untrusted) analyzers, e.g., insurance companies and academic institutions, for facilitating data-mining purposes. Meanwhile, it also hopes to keep individual whereabouts private from the analyzers. To this end, the data curator employs a location privacy protection algorithm to obfuscate actual trajectories D_A, resulting in an offline dataset of obfuscated trajectories D_O. After that, D_O is published to the data analyzers.

Design objectives. The data curator expects that the location privacy protection algorithm should achieve the following two objectives. (i) Simultaneous preservation of geographical utility as well as semantic utility. In other words, the obfuscated locations should be close to the actual locations and have similar semantics to the actual location as much as possible. (ii) Effective prevention from the location inference attacks and the de-anonymization attacks. In other words, the chance that any actual location or individual identity is derived by observing the released dataset D_O should be restricted to some magnitude specified by the data curator.

The location privacy protection algorithm, denoted by f: $D_A \to D_O$, is a randomized function that outputs the obfuscated trajectory dataset D_O given the actual trajectory dataset D_A. In this work, the f is implemented by UDPT, a privacy-protecting and utility-optimized trajectory synthesis algorithm. $P(f)$ measures the privacy security of the algorithm. The obfuscation usually leads to a loss of data utility, denoted by $Q(f)$. The computation of $P(f)$ and $Q(f)$ depends on specified metrics, which will be elaborated in our experimental Section 5.

3.2. Differential Privacy

Stemming from the area of statistical disclosure control, differential privacy [7] has become a widely accepted privacy standard. In general, differential privacy requires that the outcome of any query to a dataset is insensitive to the change (e.g., addition and removal) of a single record in that dataset. The formal definition of differential privacy is given as follows.

Definition 1 (ϵ-differential privacy)**.** *A privacy protection algorithm f can provide ϵ-differential privacy if and only if any two neighboring datasets D_1 and D_2 differing on at most one record and for any possible output $O \in Range(f)$:*

$$\Pr(f(D_1) = O) \leqslant e^{\epsilon} \times \Pr(f(D_2) = O) \tag{1}$$

where the possibility is taken over the randomness of f, $Range(f)$ denotes the set of possible outputs of the mechanism, and ϵ is called the privacy budget.

In this work, f is the trajectory synthesis algorithm, UDPT, which takes an actual trajectory dataset D_A as input and outputs an obfuscated trajectory dataset D_O, i.e., $f(D_A) = D_O$. The any two neighboring databases D_1, D_2 are expressed as any two trajectory datasets D_A, D'_A differing on at most one location record $l_t \in D_A$, i.e., $(D_A - D'_A) \bigcap (D'_A - D_A) = \{l_t\}$. The $Range(f)$ means the set of all possible outputs of f, and O the any possible output in $Range(f)$. According to Equation (1), when the output of UDPT is insensitive to the change of any single location record in the actual trajectory dataset, we say that UDPT is ϵ-differentially private. In other words, untrusted analyzers cannot derive the existence of any single location record, including the record on sensitive locations, e.g., home and hospital, in the actual dataset D_A by observing the obfuscated trajectory dataset D_O, so individual location privacy is protected.

In addition, ϵ is privacy budget of the data curator. Generally, a smaller ϵ leads to larger randomness, which further results in a stronger privacy guarantee while a poor preservation of data utility. Therefore, ϵ can be used to tune the trade-off between privacy and data utility of UDPT.

3.3. Mechanisms and Properties of Differential Privacy

Note that the differential privacy is actually a goal of privacy protection, rather than a specific privacy protection algorithm. To build a specific privacy protection algorithm, e.g., the trajectory synthesis algorithm in this work, the differential privacy expresses the algorithm's access to the dataset as a series of queries on that dataset. For instance, the private location clustering of UDPT, see Section 4.2, is essentially a series of queries on the actual trajectory dataset with the objective function. Individual privacy is protected by injecting random noise into the query results. There are mainly two types of queries, namely, numeric queries and categorical queries. Previous studies have provided two general-purpose and differentially private algorithms, namely Laplace mechanism [7] and exponential mechanism [24], to generate random noise for a numeric query and a categorical query, respectively. We introduce the two mechanisms as follows.

Laplace mechanism. Dwork et al. [7] proposed this mechanism which takes as inputs a dataset D, a privacy protection algorithm f, and the privacy budget ϵ. Theorem 1 presents a formal definition It is designed for answering the numeric query whose output is real. It added a Laplace noise to the actual query result $f(D)$. The Laplace noise is sampled from Laplace distribution $Lap(b)$ with the probability density function $Pr(x|\mu, b) = (2b)^{-1}e^{-b^{-1}|x-\mu|}$, where μ is the location parameter and b is the scale parameter determined by sensitivity Δf and desired privacy budget ϵ : $b = \Delta f/\epsilon$. μ is usually assigned 0. The sensitivity Δf indicates the maximum change of the actual query result $f(D)$ when removing or adding one record from the dataset D, which relies on the specific query.

Theorem 1. *For any privacy protection algorithm $f : D \to R^d$, the following algorithm achieves ϵ-differential privacy.*

$$A(D) = f(D) + Lap(\Delta f/\epsilon) \tag{2}$$

Exponential mechanism. For the query whose output is not real , i.e., the domain of output is categorical, McSherry et al. [24] proposed the exponential mechanism. Theorem 2 presents a formal definition of the exponential mechanism. It defines a score function q that assigns a real-valued utility score to each categorical output $r \in R$. The exponential mechanism assigns exponentially higher probabilities of being selected to outputs of higher utility scores. ϵ represents the privacy budget. The sensitivity of the score function Δq equals to the maximum change of the score function q when removing or adding one record from the dataset D, which relies on the specific score function that we use.

Theorem 2. *For any privacy protection algorithm $f : D \to R$, choosing an output $r \in R$ with probability proportional to $e^{\frac{\epsilon q(D,r)}{2\Delta q}}$ ensures the algorithm ϵ-differential privacy [24].*

The properties of differential privacy define the composability between differentially private algorithms, e.g., the Laplace mechanism and the exponential mechanism, and enables the building of a complex algorithm. Given some differentially private algorithms $f_1, f_2, \ldots, f_n$ that satisfies $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$-differential privacy respectively, the properties of differential privacy also describe the relationship between the total privacy budget (indicating the overall privacy guarantee) and the privacy budget pieces of sub-algorithms. We introduce the two properties as follows.

Theorem 3 (sequential composition [24])**.** *Let the privacy protection algorithm f_i each provide ϵ_i-differential privacy, $1 \leq i \leq n$, then running in sequence all algorithms over a dataset D provides $\sum_i \epsilon_i$-differential privacy.*

The sequential composition indicates that a sequence of algorithms that each provides differential privacy in isolation also provides differential privacy, but the privacy budget is accumulated. In other words, the sequential execution of these mechanisms on a dataset consumes $\sum_{i=1}^{n} \epsilon_i$ budget.

Theorem 4 (parallel composition [24])**.** *Let the privacy protection algorithm f_i each provide ϵ_i-differential privacy, $1 \leq i \leq n$, then applying each algorithm over a set of disjoint datasets D_i provides $max\{\epsilon_i\}$-differential privacy.*

The parallel composition means that if the sequence of privacy protection algorithms is performed on disjoint datasets, the privacy budget is determined by the largest one of all algorithms, i.e., $max(\epsilon_i)$, $1 \leq i \leq n$.

In particular, post-processing the output of a differentially private algorithm does not deteriorate the privacy, e.g., exploiting that output as an input to another algorithm or even publicly releasing that output does not violate differential privacy [24].

4. Trajectory Synthesis

In this section, we describe the procedure of the trajectory synthesis with UDPT. First, we provide an overview of the UDPT. Then, we elaborate on each phase.

4.1. Overview

Overall ideas. Recall the design objectives in Section 3.1. (i) One of the design objectives is to achieve that the chance that any actual location or individual identity is derived by observing the released dataset should be restricted to some magnitude specified by the data curator, we employ the differential privacy as the privacy notion of UDPT. An ϵ-differentially private algorithm can ensure that the output is insensitive to the change of any location record in the actual trajectory, so the location inference

attacks on actual locations can be thwarted. The privacy guarantee can be tuned by the data curator with the privacy budget ϵ. However, the differential privacy cannot prevent the de-anonymization attacks on individual identity [6]. Our remedy is to synthesize a number of indistinguishable and obfuscated trajectories for each actual trajectory in a generative manner, so the attacker cannot rebuild the linkage between individual identities and obfuscated trajectories. (ii) The other objective is to achieve simultaneous preservation of geographical utility as well as semantic utility. In other words, the obfuscated locations should be close to the actual locations and have similar semantics to the actual location as much as possible. To this end, we model the preservation of both types of utility as a multiple-objective optimization problem.

In general, UDPT is a location privacy protection algorithm, which takes as input an actual trajectory dataset of individuals, together with a (total) privacy budget ϵ, and then produces an obfuscated trajectory dataset. It enables the trajectory data curator to release an offline trajectory dataset to public or other analyzers for data-analysis purposes in a privacy-protecting and utility-preserving fashion. Although the privacy protection will obfuscate the actual data and thus deteriorate the data utility, the data curator can control the trade-off between privacy and data utility by tuning the (total) privacy budget ϵ to achieve the requirements of individuals and data analyzers. The pseudocode in Algorithm 1 presents a skeleton of UDPT. We also provide an illustrative example in Figure 2.

Algorithm 1 Utility-optimized and differentially private trajectory synthesis (UDPT).

Input: actual trajectory dataset D_A, (total) privacy budget ϵ.
Output: obfuscated trajectory dataset D_O.

1: Divide the total privacy budget ϵ into 3 pieces, namely, ϵ_1, ϵ_2, and ϵ_3.
2: Let clusters $\{C_1, C_2, \dots\} \longleftarrow$ Private location clustering with Algorithm 2, D_A and ϵ_1.
3: Let $D_O \longleftarrow \emptyset$.
4: **for each** $\boldsymbol{l}$ in D_A **do**
5: **for each** $l_t \in \boldsymbol{l}$ **do**
6: Let $L_t^{(opt)} \longleftarrow$ Privately selecting utility-optimized candidate obfuscated locations for l_t with Algorithm 3, clusters $\{C_1, C_2, \dots\}$, and ϵ_2.
7: **end for**
8: Let $\boldsymbol{L}^{(opt)} \longleftarrow (L_1^{(opt)}, L_2^{(opt)}, \dots)$.
9: Let $\boldsymbol{L}_o \longleftarrow$ Privately select obfuscated trajectories with Algorithm 4, $\boldsymbol{L}^{(opt)}$, and ϵ_3.
10: Let $D_O \longleftarrow D_O \cup \boldsymbol{L}_o$.
11: **end for**
12: **return** D_O.

UDPT is composed of three sequential phases, where each phase is a differentially private sub-algorithm. In line 1, we divide the (total) privacy budget into three pieces and each phase (or sub-algorithm) consumes one piece.

In the first phase, as shown in line 2, in order to defend against the location inference attacks, locations that occur in the actual trajectory dataset D_A are blurred into clusters by the private location clustering algorithm 2. Since this sub-algorithm is essentially iteratively updating a numeric objective function, UDPT exploits the Laplace mechanism to ensure ϵ_1-differentially private. The sub-algorithm finally produces a set of clusters $\{C_1, C_2, \dots\}$.

In the second phase, as given by line 3 to 8, to simultaneously preserve the geographical utility as well as the semantic utility of each actual trajectory $\boldsymbol{l} \in D_A$, UDPT privately selects a set of utility-optimized candidate obfuscated locations from the clusters $\{C_1, C_2, \dots\}$ for each actual location $l_t \in \boldsymbol{l}$. The selection is modeled as a multi-objective optimization problem and then solved by a differentially private genetic algorithm. Since this sub-algorithm is essentially a query which takes as input the clusters and the actual location l_t and then produces categorical outcomes, i.e., a number of location, UDPT employs the exponential mechanism to ensure the sub-algorithm ϵ_2-differentially private. After that, a sequence of utility-optimized candidate obfuscated location sets $\boldsymbol{L}^{(opt)}$ for each actual trajectory $\boldsymbol{l}$ is produced, as shown in line 8.

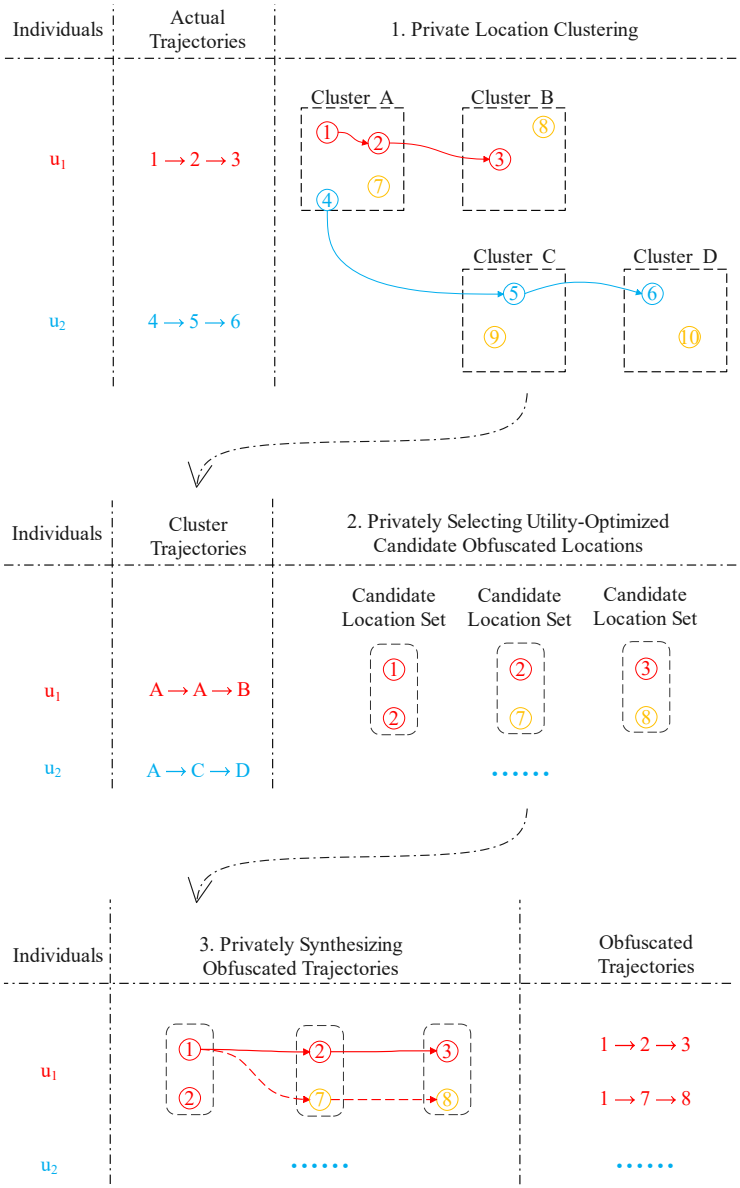

Figure 2. An example of releasing differentially private and utility optimized trajectories with UDPT. We focus on two moving individuals, namely u_1 and u_2, represented by red and blue colors, respectively. A location is denoted by a number. The yellow numbers represent the locations visited by other individuals.

In the third phase, as shown in line 9, to defend against the de-anonymization attacks, UDPT privately selects obfuscated trajectories with the Algorithm 4 and $\boldsymbol{L}^{(opt)}$. Because the sub-algorithm is essentially a categorical query which takes as input the actual trajectory $\boldsymbol{l}$ and then produces discrete outcomes, UDPT exploits the exponential mechanism to ensure ϵ_3-differentially private. After that, the sub-algorithm produces a number of indistinguishable and obfuscated trajectories $\boldsymbol{L}_o$, which share the most similar movement patterns to the actual trajectory $\boldsymbol{l}$, so $\boldsymbol{l}$ can be hidden from the de-anonymization attackers. In the end, UDPT collects all the obfuscated trajectories into the dataset D_O, as given by line 10.

Finally, the Algorithm 1 returns an obfuscated trajectory dataset D_O. The data curator can release it to public or any other data analyzers for data-mining purposes. We elaborate on each phase (or sub-algorithm) as follows.

4.2. Private Location Clustering

Since the location inference attacks aim to infer individuals' actual (or precise) locations from obfuscated locations, a widely used remedy is to replace the actual locations with blurred locations such as grids [6] or clusters [22]. The location privacy protection algorithms based on grids split the geographical space where individuals move into uniform and rectangular cells. The locations inside a cell are represented by the cell itself. However, a careless setting of the cell size is likely to divide a place in nature into different cells, which could lose original location semantics. In contrast, recent works have demonstrated that the clustering has a more natural division on geographical space, so the neighboring locations that share the same semantics have a greater chance to gather together [22]. Therefore, despite defending against location inference attacks, another advantage of location clustering is the better preservation of the semantic utility of individual trajectories. We apply K-means as the fundamental clustering algorithm. It is easy to be extended for privacy protection, especially for differential privacy, because of its efficiency in computing and simplicity in parameter configuration. However, previous efforts showed that clustering on individual data without any privacy protection measures could lead to a compromise on privacy, e.g., the adversary could find out whether a given location is in a certain cluster [25]. Therefore, we need to propose a differentially private location clustering algorithm. Despite the clustering, a recent study [26] on private classification provides us with another inspiration. We leave it as our future work.

The basic idea of location clustering is calculating the distance of pairwise (actual) locations and then gathering those locations having shorter distances iteratively. To ensure the clustering is differentially private, the random noise generated by the Laplace mechanism is injected into the objective function of K-means clustering. Finally, actual (or original) locations are blurred into clusters, then each actual trajectory is translated into a sequence of clusters, i.e., a cluster trajectory. The pseudocode of private location clustering is presented in Algorithm 2.

Algorithm 2 Private location clustering.

Input: geographical space $\mathcal{L}$, number of clusters K, privacy budget ϵ_1, max number of iterations p, sensitivity Δ.
Output: a set of clusters $\{C_1, C_2, \ldots, C_K\}$.

Initialize cluster centroids $c_1, c_2, \ldots, c_K \in \mathcal{L}$ randomly.
for $r = 1$ to p **do**
  Initialize each cluster C_k with $\emptyset$, $k = 1, 2, \ldots, K$.
  // Assign the locations in $\mathcal{L}$ to their closest cluster centroid.
  for $i = 1$ to $|\mathcal{L}|$ **do**
    $k^* := \arg\min_{k=1,2,\ldots,K} \|l_i - c_k\|_2$, $l_i \in \mathcal{L}$.
    Add l_i into the cluster C_{k^*}.
    Let $\gamma_{i,k^*} \longleftarrow 1$.
    for $k \in \{1, 2, \ldots, K\} - \{k^*\}$ **do**
      Let $\gamma_{i,k} \longleftarrow 0$.
    end for
  end for
  Calculate the objective function $g = \sum_{i=1}^{|\mathcal{L}|} \sum_{k=1}^{K} \gamma_{i,k} \|l_i - c_k\|_2 + Lap(\frac{p\Delta}{\epsilon_1})$.
  if the objective function g converges, **then**
    Break.
  end if
  // Update each cluster centroid with the mean of all locations in that cluster.
  for $k = 1$ to K **do**
    Let $c_k \longleftarrow \frac{1}{|C_k|} \sum_{l \in C_k} l$.
  end for
end for
return $\{C_1, C_2, \ldots, C_K\}$.

In line 1, we initialize K cluster centroids with the locations uniformly chosen from the geographical space $\mathcal{L}$ at random, where $\mathcal{L}$ is actually the set of locations occurring in the actual trajectory dataset D_A. After that, we iteratively update the clusters until the convergence. The steps in a single iteration are elaborated as follows.

In line 3, we initialize each cluster with an empty set. In line 5 to 12, we assign each location in the geographical space $\mathcal{L}$ to their closest cluster centroid. Specifically, first, in line 6, given a location $l_i \in \mathcal{L}$, let k^* denote the index of the cluster centroid which is closest to l_i. In particular, if there were more than one closest centroids, we uniformly chose one from them at random. Next, we add the location l_i into the cluster C_{k^*}, as shown in line 7. Then, to facilitate the calculation of objective function, we introduce an indicator of the relation between locations and clusters, which is defined in Equation (3). Let $\gamma_{i,k} = 1$ indicate the fact that the location l_i belongs to the cluster C_k, and $\gamma_{i,k} = 0$ otherwise. In this case, we have $\gamma_{i,k^*} = 1$, as shown in line 8. We also have $\gamma_{i,k} = 0$ for $k \neq k^*$, as shown in line 10.

$$\gamma_{i,k} = \begin{cases} 1, & \text{if } l_i \in C_k \\ 0, & \text{otherwise .} \end{cases} \tag{3}$$

In line 13, we calculate the objective function of the private location clustering, g, which is defined in Equation (4), K the number of clusters, l_i the i-th location in $\mathcal{L}$, c_k the k-th cluster centroid, $Lap(\frac{p\Delta g}{\epsilon_1})$ the random Laplace noise generated by the Laplace mechanism in Theorem 1.

$$g = \sum_{i=1}^{|\mathcal{L}|} \sum_{k=1}^{K} \gamma_{i,j} \|l_i - c_k\|_2 + Lap(\frac{p\Delta g}{\epsilon_1}) \tag{4}$$

In Equation (4), the maximum number of iterations, p, is a hyperparameter of the algorithm. A larger p theoretically results in a better performance of clustering. However, since we need to divide the privacy budget ϵ_1 into p pieces, a larger number of iterations could inject larger noise to a single iteration. The iteration often ends in advance, i.e., the actual number of iterations is likely less than p, so we can leave the unused privacy budget to the following steps including the Sections 4.3 and 4.4. We will discuss this hyperparameter in our experiments.

In addition, in Equation (4), ϵ_1 represents the privacy budget allocated to the private location clustering algorithm. It controls the trade-off between privacy and data utility, where a smaller value indicates a stronger guarantee of privacy while a poor preservation of data utility. Previous studies suggest that taking values in the interval $[0.1, 10]$ often brings about a reasonable trade-off between privacy and data utility [7].

In Equation (4), the Laplace noise, $Lap(\frac{p\Delta g}{\epsilon_1})$, is actually a random variable following the Laplace distribution, a.k.a., double exponential distribution. Its probability density function is defined in Equation (5), where x represents a possible noise value, $pdf(x)$ means the probability density w.r.t. x, and Δg is the sensitivity of the objective function. According to the Laplace mechanism [7], the sensitivity indicates the maximum change of $\sum_{i=1}^{|\mathcal{L}|} \sum_{k=1}^{K} \gamma_{i,j} \|l_i - c_k\|_2$ when removing or adding one location record from the actual trajectory dataset D_A. Therefore, we define the sensitivity Δg as the maximum distance between locations in the geographical space $\mathcal{L}$; that is, $\Delta g = max\{\|l_i - l_j\|_2 \mid \forall\, l_i, l_j \in \mathcal{L}\}$. In this case, we can ensure that the output of the private location clustering algorithm is insensitive to the change of a single record in the actual dataset. In other words, we achieve the differential privacy for our private location clustering algorithm.

$$pdf(x) = \frac{\epsilon_1}{2p\Delta g} \exp(-\frac{\epsilon_1 |x|}{p\Delta g}),\ x \in \mathbb{R} \tag{5}$$

In line 14, we check whether the objective function g converges. If the change of objective function values between consecutive iterations is not greater than some threshold, we say that the objective function converges. The threshold is a hyperparameter that needs

to be empirically determined. See the Section 5 and the Appendix A for a detailed discussion. When g converges, we terminate the iteration and return the clusters $\{C_1, C_2, \ldots, C_K\}$. Otherwise, in line 19, we update each cluster centroid c_k with the mean of all locations in that cluster, $k = 1, 2, \ldots, K$.

4.3. Privately Selecting Utility-Optimized Candidate Obfuscated Locations

After private location clustering, the actual trajectory is translated into a sequence of clusters, i.e., a cluster trajectory. Since most of the data utility of the actual trajectory, e.g., location transition patterns [20], is still maintained, an intuitive way to synthesize the obfuscated trajectory is to sample a trajectory, which preserves the data utility of the actual trajectory to the most extent, from the Cartesian product of the clusters at all time instants of the cluster trajectory. However, the number of candidate obfuscated trajectories is so large that we cannot find the expected trajectory at an acceptable time cost. Alternatively, a feasible remedy is to prune the sample space by shrinking each cluster to a small set of candidate obfuscated locations maintaining the data utility of the actual location as much as possible.

In this subsection, we select utility-optimized candidate obfuscated locations for each actual location in the actual trajectory. Specifically, first, we model the simultaneous preservation of both types of data utility as a multi-objective utility optimization problem. Second, we solve the optimization problem with the genetic algorithm to select utility-optimized candidate obfuscated locations, where the selection is perturbed by differential privacy to prevent the actual locations from being deduced by observing the candidate obfuscated locations.

4.3.1. Modeling Utility Optimization Problem

Given any actual location l_t in the actual trajectory $\boldsymbol{l}$, there are two objectives for selecting candidate obfuscated location, namely, (i) preserving the geographical utility of the actual location as much as possible, and (ii) maintaining the semantic utility of the actual location to the most extent. On the one hand, in order to preserve the geographical utility, a solution that has been proved feasible by existing work [12] is to select the location nearby the actual location as the obfuscated location. In this work, we borrow this solution. On the other hand, to preserve the semantic utility, we prefer to select the locations sharing the same (or similar) semantics with the actual location as the obfuscated location.

However, there exist conflicts between the two objectives since nearby locations do not always have similar semantics [10]. Therefore, an optimal decision needs to be taken in the presence of trade-offs between the two conflicting objectives, which is known as a multi-objective optimization problem. Generally, the way to solve the problem is formalizing the objectives as functions and then finding the optimal solution over the objective functions by optimization algorithms such as genetic algorithm. We formalize the two objectives as follows.

The first objective, i.e., (i) preserving the geographical utility of the actual location as much as possible, is defined as selecting M candidate obfuscated locations, denoted by $L_t^{(geo)}$, which are closest to the actual location l_t. Equation (6) presents a formal expression, which means minimizing the total Euclidean distance between M candidate obfuscated locations and the actual location l_t.

$$L_t^{(geo)} = arg\ min \sum_{i=1, l_i \in \mathcal{L}}^{M} \|l_t - l_i\|_2 \tag{6}$$

Before formalizing the second objective, we need to acquire the location semantics and define the semantic distance measure. The acquirement of location semantics can be achieved by the application programming interfaces (APIs) of many LBS providers, such as Foursquare and Google Maps. Foursquare also provides a hierarchical tree representation of location categories [27], which contains ten coarse-grained categories in the first level

and hundreds of fine-grained ones in the second level. The path distance between two category nodes in the tree indicates their discrepancy in semantics, which inspires us to propose a graph-theory-based distance metric to measure the discrepancy between the categories of the candidate obfuscated location l_c and the actual location l_t. We denote the semantic distance by $dist_{sem}(\cdot,\cdot)$ and give its definition by Equation (7).

$$dist_{sem}(l_t, l_c) = \frac{d(l_t, l_c)}{d(root, l_t) + d(root, l_c)} \tag{7}$$

where $d(\cdot,\cdot)$ denotes the length of the shortest path between the categories of two locations on the semantic tree, *root* represents the root node of the tree. In particular, if two categories are equal, their semantic distance is 0, while if they have the same parent category, the distance is 2. The semantic difference between two categories will be normalized by the sum of the nodes' depths; that is, the distance to the root node.

The second objective, i.e., (ii) maintaining the semantic utility of the actual location to the most extent, is defined as selecting M candidate obfuscated locations, denoted by $L_t^{(sem)}$, which are most similar to the actual location l in terms of location semantics. Then, the second objective is formulated as minimizing the semantic distances between the candidate obfuscated locations and the actual location, which is given by Equation (8).

$$L_t^{(sem)} = arg\ min \sum_{i=1, l_i \in C}^{M} dist_{sem}(l_t, l_i) \tag{8}$$

Combining the above two objectives, the multi-objective utility optimization problem is modeled as Equation (9) , where $L_t^{(opt)}$ represents the set of utility-optimized candidate obfuscated locations corresponding to the actual location l_t.

$$L_t^{(opt)} = arg\ min\{ \sum_{i=1, l_i \in \mathcal{L}}^{M} \|l_t - l_i\|, \sum_{i=1, l_i \in C}^{M} dist_{sem}(l_t, l_i)\} \tag{9}$$

4.3.2. Privately Selecting Candidate Obfuscated Locations

The solutions of the multi-objective optimization problem include particle swarm, ant colony, and genetic algorithm [28]. Among these approaches, the genetic algorithm is well-known for its high performance. It is based on a natural selection process that mimics biological evolution. Recent work has demonstrated that accessing actual data without taking any privacy-protection measures could lead to compromise on individual privacy, so Zhang et al. [29] proposed a differentially private genetic algorithm. However, their work is not designed for high-dimensional data, such as trajectory, so we made an adaption, as follows, to support trajectory data.

According to the multi-objective utility optimization problem defined in Equation (9), we define the objective function of the genetic algorithm, given by Equation (10), where α indicates the data curator's preference on the geographical utility and $1-\alpha$ the semantic utility. For example, suppose that the released data were used for early LBS that relied more on the geographical utility, e.g., points-of-interests extraction [30], then α should be greater than $1-\alpha$, namely, $\alpha \in (0.5, 1.0)$. The geographical distance and semantic distance in the equation are shifted and re-scaled so that they end up having the same range, $[0, 1]$.

$$q(L_t^{(opt)}) = \alpha \sum_{i=1, l_i \in L_t^{(opt)}}^{M} dist_{sem}(l_t, l_i) + (1-\alpha) \sum_{i=1, l_i \in L_t^{(opt)}}^{M} dist_{sem}(l_t, l_i) \tag{10}$$

In addition, in the genetic algorithm, there exist four operations, namely, encoding, selection, crossover, and mutation. (i) The encoding operation represents the solution of the optimization problem by a location set. (ii) The selection operation randomly chooses one or more location sets that maximize the objective function from an intermediate set.

(iii) The crossover operation randomly exchanges some locations in a location set with the counterpart in another location set. (iv) The mutation operation randomly replaces a location in a location set by another location. Note that the selection operation involves access to actual (private) data, thus it should be perturbed by random noise to ensure differentially private. In contrast, the other three operations only access the perturbed results, so no extra perturbation is required according to the post-processing property of differential privacy in Section 3. We formulate the procedure of privately selecting utility-optimized candidate obfuscated location set by Algorithm 3.

Algorithm 3 Privately selecting utility-optimized candidate obfuscated locations.

Input: actual location l_t, privacy budget ϵ_2, size of candidate obfuscated location set M, number of intermediate sets m, number of selected sets m', number of iterations r, the cluster C containing l_t

Output: candidate obfuscated location set $L_t^{(opt)}$

1: // encoding operation
2: Initialize the intermediate set Ω with m candidate obfuscated location sets randomly sampled from the cluster C, where the size of each set is M.
3: **for** $i = 1$ to $r - 1$ **do**
4: // selection operation
5: Initialize the selected set $\Omega' = \emptyset$.
6: **for each** $L_t^{(opt)} \in \Omega$ **do**
7: Compute $\Pr(L_t^{(opt)}) = \frac{\exp(\frac{\epsilon_2 q(L_t^{(opt)})}{2r\Delta q})}{\sum_{L_t^{(opt)'} \in \Omega} \exp(\frac{\epsilon_2 q(L_t^{(opt)})}{2r\Delta q})}$
8: **end for**
9: Randomly sample m sets from Ω following $\Pr(L_t^{(opt)})$ and put them into Ω'.
10: Set $\Omega = \emptyset$.
11: **for** $j = 1$ to $m'/2$ **do**
12: Randomly select two sets $L_t^{(opt)'}, L_t^{(opt)''} \in \Omega'$.
13: // crossover operation
14: Randomly crossover $L_t^{(opt)'}$ and $L_t^{(opt)''}$.
15: // mutation operation
16: Randomly mutate $L_t^{(opt)'}$ and $L_t^{(opt)''}$.
17: Add $L_t^{(opt)'}$ and $L_t^{(opt)''}$ into Ω.
18: **end for**
19: **end for**
20: **for each** $L_t^{(opt)} \in \Omega$ **do**
21: Compute $\Pr(L_t^{(opt)}) = \frac{\exp(\frac{\epsilon_2 q(L_t^{(opt)})}{2r\Delta q})}{\sum_{L_t^{(opt)'} \in \Omega} \exp(\frac{\epsilon_2 q(L_t^{(opt)})}{2r\Delta q})}$
22: **end for**
23: Randomly select a set $L_t^{(opt)}$ following $\Pr(L_t^{(opt)})$ from Ω.
24: **return** $L_t^{(opt)}$.

In line 1 to 2, we encode the solution of the optimization problem by a location set. The intermediate location set is initialized with m sets of candidate obfuscate locations, where each set is composed of M locations randomly selected from the cluster C. The initialization of the intermediate sets provides an initial direction for the optimal solution searching of the genetic algorithm.

In lines 3 to 23, we select the utility-optimized candidate obfuscated location set in a differentially private and iterative manner. We uniformly divide the privacy budget ϵ_2 into r pieces, each iteration consumes ϵ_2/r.

In line 4, we employ the selection operation to choose m candidate obfuscated location sets that preserve the geographical and semantic utility to the most extent. Since the selection involves access to the actual trajectory data, we exploit the exponential mechanism to provide a differentially private guarantee, where the probability of the candidate obfuscated location set $L_t^{(opt)}$ being chosen is proportional to its objective function value, $q(L_t^{(opt)})$. In line 7, the sensitivity Δq of the exponential mechanism equals 1. In line 9, we randomly sample m location sets without replacement from the intermediate sets Ω following the probability distribution $\Pr(L_t^{(opt)})$, which can be achieved by constructing an unbalanced roulette.

In line 11, we randomly select $m'/2$ pairs of location sets from the selected set, ω', and put them into the intermediate set, Ω. To this end, first, we randomly select two location sets, $L_t^{(opt)'}$ and $L_t^{(opt)''}$, from Ω'. Next, we crossover these two location sets by randomly exchanging part of locations in $L_t^{(opt)'}$ with the counterparts in $L_t^{(opt)''}$. Then, we randomly mutate the location set $L_t^{(opt)'}$ (and $L_t^{(opt)''}$) by replacing a location in $L_t^{(opt)'}$ (and $L_t^{(opt)''}$) by another location in the cluster C. Finally, we put $L_t^{(opt)'}$ and $L_t^{(opt)''}$ into the intermediate set Ω.

In line 20, the iteration ends. We randomly choose a candidate obfuscated location set from the intermediate set, Ω, following the probability distribution $\Pr(L_t^{(opt)})$ as the algorithm output. The relation between m and m' is determined empirically. $10m \leq m' \leq 20m$ often results in good performance.

We generate the candidate obfuscated location set, $L_t^{(opt)}$, for the actual location at each time instant, l_t, of the actual trajectory $\boldsymbol{l}$ with Algorithm 3. Then, we obtain a sequence of candidate obfuscated location sets corresponding to $\boldsymbol{l}$, denoted by $\boldsymbol{L}^{(opt)} = (L_1^{(opt)}, L_2^{(opt)}, \ldots, L_t^{(opt)}, \ldots)$, which will be used for the synthesis of obfuscated trajectories of $\boldsymbol{l}$. We do the same for each actual trajectory $\boldsymbol{l}$ in D_A.

4.4. Privately Synthesizing Obfuscated Trajectories

A common idea of existing trajectory synthesis methods is to build a trajectory generator that has learned the population movement patterns and then produce the obfuscated trajectory dataset D_O in a generative manner. However, individual movement patterns could be destroyed, so numerous data analysis tasks that rely on individual-level data utility, e.g., periodic patterns mining [9] could suffer poor performance. To tackle this issue, we synthesize the obfuscated trajectories for each individual, i.e., each actual trajectory, independently. In this case, the privacy budgets consumed by all individuals do not accumulate according to the differential privacy's parallel composition property. In other words, when the total privacy budget is fixed, we can spare a larger privacy budget for the trajectory synthesis of a single individual, resulting in less noise injection and thus better preservation of data utility. In addition, to defend against the de-anonymization attacks, for each actual trajectory, we privately synthesize more than one indistinguishably obfuscated trajectories that share the most similar data utility with the actual trajectory.

In the previous section, for any actual trajectory $\boldsymbol{l}$, we obtained a sequence of candidate obfuscated location sets. To privately synthesize obfuscated trajectories for $\boldsymbol{l}$, first, to preserve the geographical and semantic utility, we capture individual movement patterns by the CRF. Second, we synthesize a set of candidate obfuscated trajectories by the CRF sequence decoding based on the sequence of candidate obfuscated location sets. Then, some candidates are privately selected as the (final) obfuscated trajectories. Since the obfuscated trajectories maintain similar data utility with the actual trajectory, they cannot be distinguished from each other, providing a defense against de-anonymization attacks. We formalize the aforementioned steps by the pseudocode in Algorithm 4.

Algorithm 4 Private synthesizing obfuscated trajectories.

Input: sequence of candidate obfuscated location sets $\boldsymbol{L}^{(opt)}$, actual trajectory $\boldsymbol{l}$, sensitivity $\Delta \Pr$, number of (final) obfuscated trajectories N.
Output: a set of (final) obfuscated trajectories $\boldsymbol{L}_o$.

1: Construct the obfuscated trajectory synthesizer with $\boldsymbol{l}$.
2: // Produce a number of candidate obfuscated trajectories.
3: $\boldsymbol{L}_c = \emptyset$.
4: **for** $i = 1$ to N **do**
5: $< \boldsymbol{l}_c^{(i)}, \Pr(\boldsymbol{l}_c^{(i)}|\boldsymbol{l}) > \longleftarrow$ Produce the i-th most probable candidate obfuscated trajectory and its probability with Viterbi, $\boldsymbol{L}^{(opt)}$ and the synthesizer.
6: Add $< \boldsymbol{l}_c^{(i)}, \Pr(\boldsymbol{l}_c^{(i)}|\boldsymbol{l}) >$ into $\boldsymbol{L}_c$.
7: **end for**
8: // Privately select the (final) obfuscated obfuscated trajectories from the candidates.
9: **for** $i = 1$ to N **do**
10: Calculate the probability of $\boldsymbol{l}_c^{(i)}$ being selected as the (final) obfuscated trajectory, i.e., $\Pr(\boldsymbol{l}_c^{(i)})$, with Equation (11).
11: **end for**
12: $\boldsymbol{L}_o \longleftarrow$ Randomly sample $\lfloor N/2 \rfloor$ candidate obfuscated trajectories from $\boldsymbol{L}_c$ following its probability distribution without replacement.
13: **return** $\boldsymbol{L}_o$.

4.4.1. Constructing Obfuscated Trajectory Synthesizer

CRF is a discriminative undirected graphical model that supports auxiliary dependency within a sequence and performs well over many problems, such as sequence inference [31]. Recent works have shown that CRF can capture individual movement patterns from spatial, temporal, and semantic aspects, even though the trajectory data are sparse [23]. In particular, CRF is good at learning the transition patterns between locations, bringing about good preservation of data utility, such as periodic movement patterns.

In this work, we consider the most important example of modeling sequences, i.e., a linear chain CRF, which models the dependency between the actual trajectory and a mobility feature sequence. We consider the following mobility features concerning each actual location l in the actual trajectory; that is, the time when the individual visits l, the day of the week when the individual visits l, the time elapsed since the previous time instant, and the category of l. We extract the mobility features for each actual location in the actual trajectory to obtain the mobility feature sequence.

As shown in line 1 of Algorithm 4, the obfuscated trajectory synthesizer is essentially the CRF trained over the actual trajectory $\boldsymbol{l}$. The way to train the CRF is following our previous work [23]. The idea is as follows. First, we split both the actual trajectory and the corresponding mobility feature sequence into a number of sub-sequences by one week to enrich the training data. Then, we train the CRF by estimating the parameters that maximize the probability of the actual trajectory conditioned on the mobility feature sequence. After the training, the individual movement patterns of the actual trajectory $\boldsymbol{l}$ have been "remembered" by the parameters of CRF.

4.4.2. Privately Selecting Obfuscated Trajectories

As given by line 2 to line 6 of Algorithm 4, with the obfuscated trajectory synthesizer, we produce the (final) obfuscated trajectories by sequence decoding of the synthesizer. The sequence decoding refers to finding the most probable trajectories corresponding to a given mobility feature sequence, which can be solved by the Viterbi algorithm. An improved version [32] can produce a number of most probable trajectories that share similar movement patterns, which inspires us to produce a number of indistinguishably obfuscated trajectories to prevent de-anonymization attacks. The trajectories produced by Viterbi are called by candidate obfuscated trajectories, represented by $\boldsymbol{L}_c$. Let N denote the number of candidate obfuscated trajectories, thus we have $\boldsymbol{L}_c = \{\boldsymbol{l}_c^{(1)}, \ldots, \boldsymbol{l}_c^{(N)}\}$. The Viterbi algorithm

can be further improved by restricting the sample space to the Cartesian product of the candidate obfuscated location sets in $\boldsymbol{L}^{(opt)}$, because the locations that cannot well preserve the data utility of the actual locations have been pruned after the utility optimization in Section 4.3.2. As shown in line 5, we exploit the improved Viterbi algorithm to produce the i-th most probable candidate obfuscated trajectory $\boldsymbol{l}_c^{(i)}$, together with its probability $\Pr(\boldsymbol{l}_c^{(i)}|\boldsymbol{l})$, where $i = 1, 2, \ldots, N$.

$\Pr(\boldsymbol{l}_c^{(i)}|\boldsymbol{l})$ represents the possibility that Viterbi regards $\boldsymbol{l}_c^{(i)}$ as the actual trajectory $\boldsymbol{l}$, which also indicates to what extent $\boldsymbol{l}_c^{(i)}$ can preserve the movement patterns in $\boldsymbol{l}$. The probability can be directly calculated by the improved Viterbi algorithm. We omit the computational details, which can be found in [32]. In line 6 of Algorithm 4, we collect the produced candidate obfuscated trajectories with the set $\boldsymbol{L}_c$ i, where $|\boldsymbol{L}_c| = N$.

The Viterbi algorithm is essentially a query that takes as input $\boldsymbol{L}^{(opt)}$ and the synthesizer, and then produces categorical outputs, i.e., a number of candidate obfuscated trajectories. Note that the above synthesizer is constructed based on the actual trajectory $\boldsymbol{l}$ without any privacy-protecting measure. In this case, the query result is sensitive to the change of a singe location in the actual trajectory $\boldsymbol{l}$. A location inference attacker could derive the actual location by observing the query result. To tackle this issue, we exploit the exponential mechanism to perturb the outputs of the Viterbi algorithm. Recall the exponential mechanism in Theorem 2, the idea is to consider the probability $\Pr(\boldsymbol{l}_c^{(i)}|\boldsymbol{l})$ as the score of the candidate obfuscated trajectory $\boldsymbol{l}_c^{(i)}$, and then to select the (final) obfuscated trajectory $\boldsymbol{l}_o^{(i)}$ from the candidates $\boldsymbol{L}_c$ with the probability proportional to $\exp(\frac{\epsilon_3 \Pr(\boldsymbol{l}_o^{(i)}|\boldsymbol{l})}{2\Delta \Pr})$. The normalized probability over all possible outputs is defined in Equation (11).

$$\Pr(\boldsymbol{l}_o^{(i)}) = \frac{\exp\left(\frac{\epsilon_3 \Pr(\boldsymbol{l}_o^{(i)}|\boldsymbol{l})}{2\Delta \Pr}\right)}{\sum_{\boldsymbol{l}_o^{(i)} \in \boldsymbol{L}_c} \exp\left(\frac{\epsilon_3 \Pr(\boldsymbol{l}_o^{(i)}|\boldsymbol{l})}{2\Delta \Pr}\right)} \tag{11}$$

where ϵ_3 is the privacy budget of the exponential mechanism, which is determined by the data curator, as shown in line 1 of Algorithm 1. $\Delta \Pr$ represents the maximum change of the score when removing or adding a single location from the actual trajectory $\boldsymbol{l}$, according to the definition of differential privacy. Since the probability $\Pr(\boldsymbol{l}_o^{(i)}|\boldsymbol{l})$ takes values in $[0, 1]$, we have $\Delta \Pr = 1$.

As show in line 8 to 12 of Algorithm 4, we randomly sample $\lfloor N/2 \rfloor$ candidates from $\boldsymbol{L}_v$ without replacement to constitute the set of final obfuscated trajectories, $\boldsymbol{L}_o$ following the probability distribution in Equation (11). The reason that we choose $\lfloor N/2 \rfloor$ (final) obfuscated trajectories from $\boldsymbol{L}_v$ is in two aspects. On the one hand, a larger number of obfuscated trajectories can increase the indistinguishability between the trajectories to prevent the de-anonymization attacks, because it is more difficult for the attacker to find the correct linkage between the numerous obfuscated trajectories and individual identities. On the other hand, a less number of obfuscated trajectories indicates that less candidate obfuscated trajectories, which have dissimilar movement patterns with the actual trajectory, are chosen as the (final) obfuscated trajectories. Therefore, $\lfloor N/2 \rfloor$ is a moderate trade-off between the above two aspects.

Let $\boldsymbol{L}_o$ denote the final obfuscated trajectory set. For each actual trajectory $\boldsymbol{l} \in D_A$, we independently select the obfuscated trajectories $\boldsymbol{L}_o$ with the same privacy budget ϵ_3. According to the parallel composition property of differential privacy in Theorem 4, the privacy budgets do not accumulate. In the end, we merge all of the obfuscated trajectory sets into a single set, i.e., the obfuscated trajectory dataset D_O.

5. Experimental Evaluations

In this section, we evaluate the utility of UDPT compared with competitors. UDPT was implemented in Python 3.8. All experiments were performed on a desktop computer

with an Intel i7 CPU and 16 GB of main memory. We ran each group of the comparative experiment 10 times and took the average. Parameter setting is listed in Appendix A.2.

5.1. Evaluation Setup

Datasets. Three real-world datasets were used for the experimental evaluations in this section. (i) The *GeoLife* [30] dataset contains a large number of moving trajectories collected from 182 mobile individuals over three years. (ii) The *Gowalla* dataset contains an undirected social network and check-ins collected by the Stanford Network Analysis Project (SNAP) from Gowalla, a popular location-based social network (LBSN), throughout 2008-2010 [33]. The social network contains friendships among individuals. A check-in (also called a record of location visit) is composed of an identity number, a location represented by the GPS latitude and longitude coordinates, the date and time when the individual visits that location. (iii) Another dataset, *Brightkite*, with a similar structure, was collected from a popular LBSN (Brightkite) [34] between 2009 and 2011.

Data preprocessing. Note that there is a broad spread of locations as well as a long period in all datasets. To avoid a dramatic increase of time and memory overhead caused by high sparsity, we took the following steps to obtain a subset from each dataset. First, we restricted our evaluations to check-ins taking place in Beijing for GeoLife, Stockholm for Gowalla, and San Francisco for Brightkite. Then, only the individuals with at least 20 check-ins, whom we considered as active individuals, were retained. We present the statistical characteristics of the datasets after preprocessing in Table 3.

Table 3. Statistical characteristics of datasets after preprocessing.

Dataset	GeoLife	Gowalla	Brightkite
City	Beijing	Stockholm	San Francisco
Time span	April 2007–August 2012	September 2009–August 2011	April 2008–October 2010
# of individuals	177	4337	474
# of check-ins	18,204,931	996,028	53,624
Average # of check-ins per individual	102,852.7	229.7	113.1

Competitors. We compare UDPT with two state-of-the-art location privacy protection algorithms. (i) *AdaTrace* is a differentially private publishing mechanism for trajectories [6]. It leverages an exponential mechanism to probabilistically merge locations based on location distances and then releases synthesized trajectories in a differentially private manner. Compared with other algorithms based on trajectory synthesis, such as [19,20], AdaTrace claimed better preservation of multiple kinds of geographical utility including query error and trajectory length, so we choose it as our competitor. (ii) *MLCE* is a mechanism combined with loss and conditional entropy [12]. It prefers choosing locations nearby actual locations as the obfuscated locations. Based on ϵ-differential privacy, MLCE adds a remapping to the mechanism's output. Since the remapping is guided by two metrics, worst-case quality loss and conditional entropy, the authors of MLCE argue that higher utility and stronger privacy guarantee can be achieved, thus we choose it as our competitor.

5.2. Privacy Analysis

We now theoretically analyze the privacy guarantee of UDPT. Since UDPT is composed of three sub-algorithms, we first analyze the privacy of each sub-algorithm and then analyze UDPT as a whole.

In Algorithm 2, the only access to the actual dataset is updating the objective function, which has been perturbed by adding Laplace noise with the privacy budget ϵ_1. Subsequent

operations do not consume any privacy budget. Consequently, according to the Theorem 1, the Algorithm 2 ensures ϵ_1-differential privacy.

Algorithm 3 has one step accessing the actual trajectory, namely, the selection operation of the genetic algorithm, which has been perturbed by the exponential mechanism. The privacy budget consumed by the mechanism is ϵ_2. We independently select candidate obfuscated locations for the actual location at each time instant of each actual trajectory in D_A. According to the parallel composition theorem in Theorem 4, the overall privacy budget consumption over D_A does not accumulate. Consequently, Algorithm 3 ensures ϵ_2-differential privacy.

Algorithm 4 also has one step accessing the actual trajectory, namely, privately selecting the (final) obfuscated trajectories, which has been perturbed by the exponential mechanism with the privacy budget ϵ_3. Since we generate obfuscated trajectories for each actual trajectory independently, the trajectory of each individual can be considered a disjointed subset of the entire trajectory dataset D_A. Thus, according to the parallel composition theorem in Theorem 4, Algorithm 4 ensures ϵ_3-differential privacy.

UDPT is a sequential combination of above three differentially private sub-algorithms, where their privacy guarantees have been proved as ϵ_1, ϵ_2, and ϵ_3, respectively. Let $\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3$. According to the sequential composition theorem in Theorem 3, we can derive that UDPT ensures ϵ-differential privacy.

5.3. Utility Metrics

We evaluate the utility $Q(f)$ in terms of semantics and geography, respectively. A larger $Q(f)$ indicates higher data utility.

5.3.1. Semantic Utility Metrics

(i) Periodic pattern Jaccard coefficient. Periodic patterns are required by numerous semantic mining tasks [9] for providing personalized services, where a periodic pattern is represented by a series of locations that repeatedly occurs in a trajectory dataset, e.g., (school, apartment, school). Given the actual dataset D_A and its obfuscated version D_O, we find out top-k periodic patterns $PP(D_A)$ and $PP(D_O)$ from each dataset, respectively. Then, the semantic utility of the private protection algorithm $Q(f)$ is evaluated by the Jaccard similarity coefficient over the two sets of periodic patterns, as shown in Equation (12). A greater coefficient indicates a higher utility in terms of semantics. Particularly, for AdaTrace, the geographical space of D_O may be different from that of D_A, $PP(D_A)$, and $PP(D_O)$ may be incomparable. To tackle this issue, trajectories in D_A and D_O are discretized to grid trajectories following [6], where the number of grids is 400.

$$Q(f) = \frac{|PP(D_A) \bigcap PP(D_O)|}{|PP(D_A) \bigcup PP(D_O)|} \tag{12}$$

(ii) Semantic annotation is a trajectory data analysis task in many emerging LBS such as location recommendation [10]. It represents the process of attaching category tags such as shopping and nightlife to locations in a trajectory dataset. A common process of semantic annotation is to train a classifier over a dataset where semantic tags are given. We represent the category tags by Foursquare location categories [27]. In our experiments, we acquire category tags for each location in the obfuscated trajectory dataset D_O from Foursquare and then split D_O into training and test subsets with a ratio of 9:1. A widely used semantic annotation approach [8] is exploited to train the classifier over the training subset and predict the semantic tags of the test subset. Specifically, first, it extracts population features and temporal features, e.g., distribution of check-in time, to express the category tag of locations in training data. Second, the relatedness among locations, a.k.a, the regularity of individual check-ins to similar locations, e.g., co-occurrence, is extracted by a related locations network. Finally, the category tags of locations in the test subset can be derived from their related locations in the training subset. Generally, the effectiveness of the semantic annotation task can be measured by *Accuracy*, which is defined as the ratio of

the number of correctly annotated locations to the total number of locations. A higher accuracy indicates better performance of the semantic annotation, which further implies higher utility in terms of semantics.

5.3.2. Geographic Utility Metrics

(i) Negative Hausdorff distance is widely used to measure the similarity of two sets of points. It is the negative of the greatest of all the distances from points in one set to the closest point in the other set. Equation (13) presents the definition of *negative Hausdorff distance* between the actual dataset D_A and obfuscated D_O:

$$Q(f) = -max\{h(D_A, D_O), h(D_O, D_A)\} \tag{13}$$

where $h(D_A, D_O) = max_{l \in D_A}\{min_{l' \in D_O}\{dist(l, l')\}\}$, $dist(\cdot, \cdot)$ is a distance measurement method, e.g., Euclidean distance. A larger $Q(f)$ indicates a higher utility in terms of geography.

(ii) Negative query error refers to the negative value of the query error, where a query is to retrieve the number of trajectories passing through a certain region R. Let q denote a query and $q(D)$ its answer on a trajectory database. The query error is defined as the ratio of the difference of query result on D_A and that on D_O to the query result on D_A. We generate 1000 random queries $\mathcal{Q}$ by randomly choosing the region from the map and taking the negative average of all query errors, as shown in Equation (14). A larger $Q(f)$ implies a higher utility in terms of geography.

$$Q(f) = -\frac{1}{|\mathcal{Q}|}\sum_{q \in \mathcal{Q}} \frac{|q(D_A) - q(D_O)|}{q(D_A)} \tag{14}$$

Note that the reason for using negative values of the query error and the Hausdorff distance is that they can vary with the semantic utility in the same direction.

5.4. Privacy Metric

Location privacy attacks metric. As we summarized in Section 2, location inference attack and de-anonymization attack are two of the most common attacks in the literature concerning location privacy protection. A common objective of both attacks [3,35] is to infer individual actual locations from obfuscated locations. Before the attack, for any location $l \in \mathcal{L}$, especially a sensitive location such as a hospital or workplace that the adversary is interested in, he or she has a prior probability distribution P_l regarding the individuals who visit l. Since the distribution represents the preference of population on l, it is assumed to be accessible to the adversary. After observing the obfuscated trajectories D_O, the adversary could derive a posterior probability distribution $P_{l|D_O}$ through the attack. It represents the preference of the individuals in D_O on l, which is assumed to be inaccessible to the adversary before the attack. The difference between $P_{l|D_O}$ and P_l indicates how much knowledge concerning the actual location l the adversary has obtained by observing the obfuscated dataset D_O. It also implies the location privacy protection algorithm's vulnerability to both the location inference attacks and the de-anonymization attacks. A larger difference indicates a weaker privacy guarantee. We measure the difference by Jensen–Shannon divergence $JS(P_l||P_{l|D_O})$, a widely used measurement of the difference between two probability distributions. 10% locations are randomly chosen from D_A uniformly at random and regarded as sensitive locations, denoted by $\mathcal{S}$. We calculate the location privacy attacks metric, $P(f)$, by taking the negative average of $JS(P_l||P_{l|D_O})$ over all sensitive locations, as given by Equation (15). A larger $P(f)$ indicates a stronger privacy guarantee.

$$P(f) = -\frac{\sum_{l \in \mathcal{S}} JS(P_l||P_{l|D_O})}{|\mathcal{S}|} \tag{15}$$

5.5. Comparative Evaluations

Three groups of empirical evaluations are conducted to compare UDPT with its competitors, which are shown in Figures 3–5, respectively. An in-depth analysis by studying the experimental results under the utility metrics one by one is shown as follows.

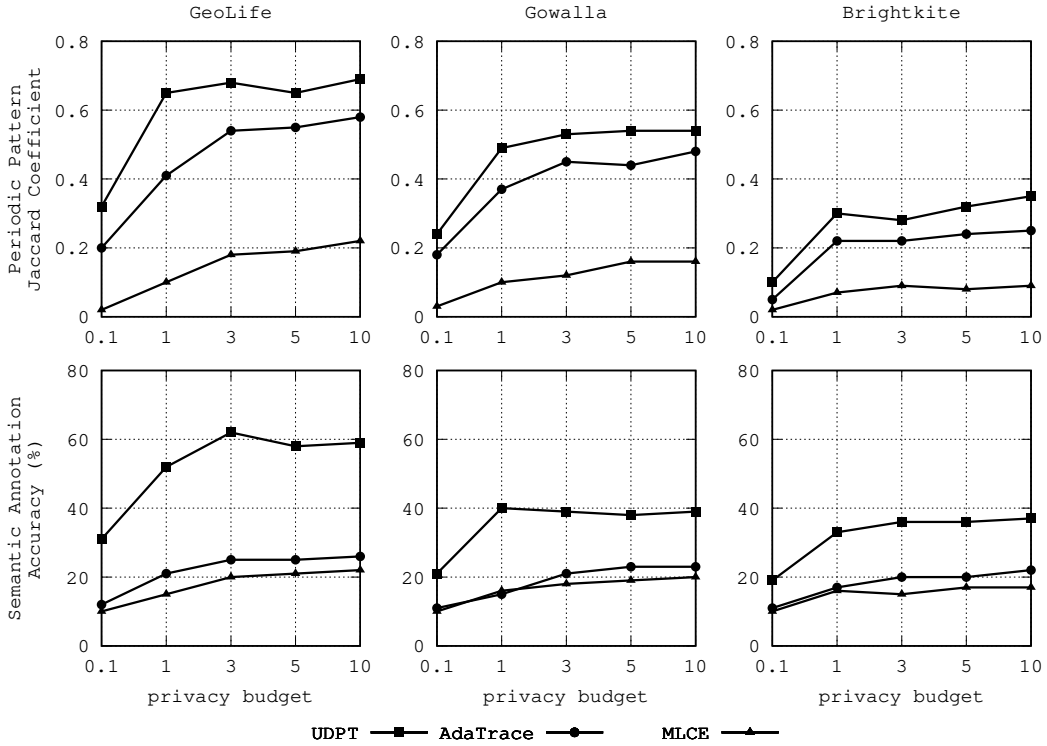

Figure 3. Comparison of UDPT against existing works in terms of semantic utility under different privacy budgets on three datasets. A higher ordinate indicates a better utility.

5.5.1. Evaluation on Semantic Utility

The semantic utility of three location privacy protection algorithms is evaluated by periodic pattern Jaccard coefficient and semantic annotation accuracy, respectively. Experimental results are shown in Figure 3. We summarize the experimental results about the semantic utility in terms of the periodic pattern Jaccard coefficient and then explain the reasons.

As shown in Figure 3, (i) we observe that UDPT always performs better than its competitors, especially in terms of semantic annotation accuracy. (ii) In most cases, the semantic utility increases when a larger privacy budget is used. (iii) In particular, in the strictest privacy setting of $\epsilon \in [0.1, 1.0]$, all of the location privacy protection algorithms provide poor performance. (iv) All of the location privacy protection algorithms provide a better semantic utility on denser trajectory datasets, such as GeoLife, while performing poorer on sparser trajectory datasets, such as Gowalla and Brightkite. Our explanations of the experimental results are as follows.

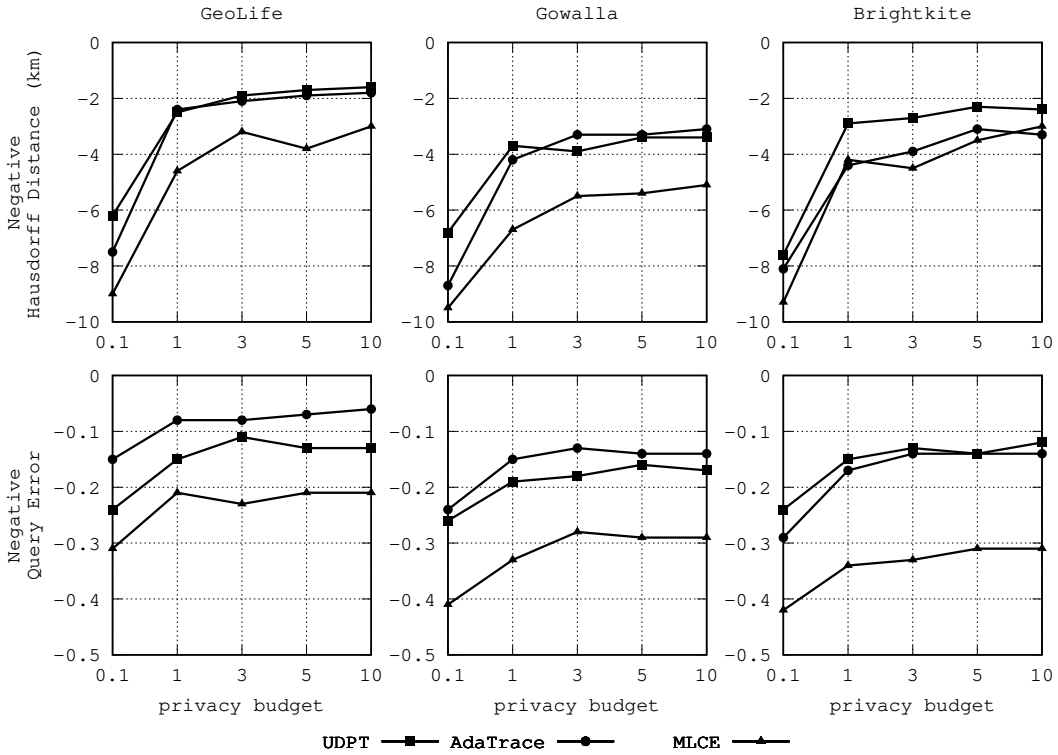

Figure 4. Comparison of UDPT against existing works in terms of geographic utility under different privacy budgets on three datasets. A higher ordinate implies a better utility.

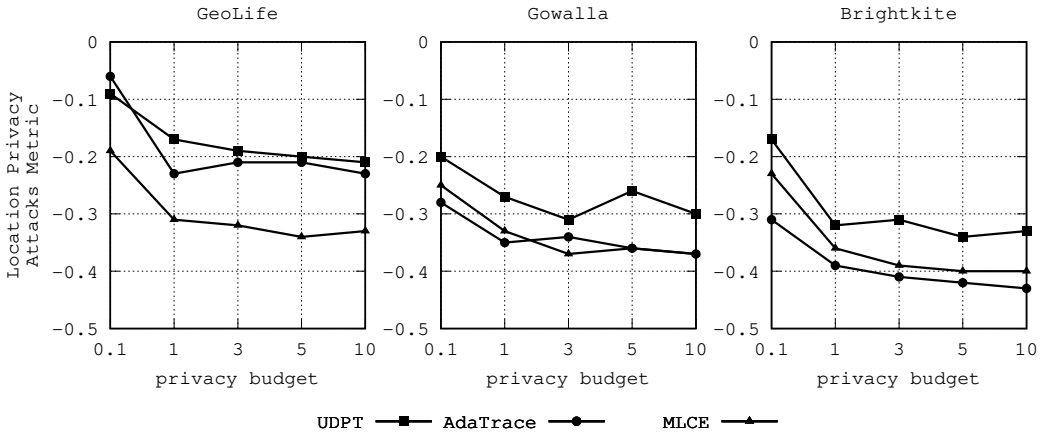

Figure 5. Comparison of UDPT against existing location privacy protection algorithms in terms of location privacy attacks metric under different privacy budgets on three datasets. A larger ordinate indicates a stronger prevention from attacks.

(i) The outperformance of UDPT against its competitors can be attributed to the good preservation of periodic patterns as well as location categories. First, the CRF model that UDPT exploits is good at capturing the movement patterns implied in the actual trajectory, especially the long-term and periodic patterns. However, AdaTrace considers only the one-step transition patterns instead of the long-term patterns of the whole trajectory. Hence, periodic patterns cannot be well preserved by AdaTrace. Similarly, MLCE also performs poorly in terms of periodic patterns since it independently obfuscates each actual location and neglects the preservation of movement patterns of the whole trajectory. In addition, the reason for UDPT's significant outperformance is that, besides the good preservation of periodic patterns, temporal features that many semantic annotation tasks rely on are also well contained. In [8], the semantic annotation tasks exploit individual mobility features

in geographical and temporal aspects, e.g, individuals' co-occurrence, visit frequency of locations, the temporal distribution of check-in time in one week and one day, to capture the relatedness among locations. However, the temporal features are destroyed by AdaTrace, so the semantic annotator could learn the wrong match between the locations and the category tags, which further leads to low accuracy of the semantic annotation. Although MLCE holds the temporal features, other mobility features such as visit frequency of locations that the semantic annotator depends on are still lost. Consequently, MLCE performs a low semantic annotation accuracy.

The result in (ii) is expected, because, according to the definition of differential privacy in Section 1, a larger budget usually brings about less noise and thus better data utility. Nonetheless, the noise will significantly increase when the privacy budget is smaller, especially for the $\epsilon \in [0.1, 1.0]$, which explains the result in (iii). The reason for (iv) is that, on the one hand, a trajectory dataset with numerous check-ins usually contains more enriched movement patterns, especially frequent and long-term patterns. In contrast, a sparse dataset, e.g., Brightkite, of which the average number of check-ins per individual is only 113, contains only sporadic visits, rather than repeated transitions between locations. Consequently, we observe that the evaluation of GeoLife performs best among all the datasets. On the other hand, a denser dataset can provide a larger volume of training data for the semantic annotator, resulting in a more in-depth capture of relatedness between locations, together with a more accurate prediction of the category tags.

5.5.2. Evaluation on Geographic Utility

We compare UDPT's geographic utility with AdaTrace and MLCE in terms of two metrics, namely, negative Hausdorff distance and negative query error.

As shown in Figure 4, (i) in general, UDPT has better geographical utility than its competitors in terms of negative Hausdorff distance. (ii) UDPT shows a poorer performance in terms of negative query error compared with AdaTrace. (iii) All location privacy protection algorithms perform better on dense datasets, such as GeoLife, than on the sparse datasets, such as Gowalla and Brightkite. The analysis of the experimental results is as follows.

(i) UDPT's outperformance, in terms of negative Hausdorff distance, is due to its preference of selecting the obfuscated locations nearby the actual locations. Although AdaTrace can also well preserve the density features, i.e., preferring to choosing nearby obfuscated locations, it still performs a little poorer than UDPT in terms of negative Hausdorff distance. The reason mainly lies in AdaTrace's randomness during selecting the obfuscated locations from the grid. In the last phase of AdaTrace, an obfuscated trajectory would be randomly synthesized based on a grid trajectory, where each obfuscated location would be uniformly chosen from a grid at random. However, if the grid was larger, AdaTrace might choose an outlier that is distributed nearby the boundary of the map as the obfuscated location. In this case, the Hausdorff distance would be larger. Consequently, AdaTrace shows a worse performance in terms of negative Hausdorff distance than UDPT. In contrast, the MLCE errors stem from the independent noise injected into the obfuscated trajectory. Compared with UDPT and AdaTrace, the locations generated by MLCE will be more widely and randomly distributed in the map, which leads to a larger distance discrepancy (or lower negative Hausdorff distance) between the actual dataset and the obfuscated one.

(ii) AdaTrace's outperformance, in terms of the negative query error, can be attributed to its good preservation of mobility features of the population. The query error implies to what extent the density features over the whole dataset are preserved. AdaTrace's density-adaptive grid can preserve the density features of the actual trajectory dataset as much as possible. In contrast, UDPT pays more attention to the utility preservation of each individual than to the population. Consequently, we observe that UDPT shows a little poorer performance in terms of negative query error than AdaTrace.

(iii) A denser trajectory dataset indicates that more places nearby the actual locations can be chosen as the obfuscated locations, resulting in better preservation of geographical utility. In addition, the map of GeoLife is smaller than those of Gowalla and Brightkite, which further increases the impact of data density on the geographical utility.

5.5.3. Evaluation on Privacy

In this subsection, we evaluate location privacy protection algorithms' privacy guarantee by the location privacy attacks metric. Experimental results are shown in Figure 5. A larger ordinate indicates less knowledge that the adversary obtains after observing the obfuscated trajectory dataset, which further implies a stronger defense against privacy attacks. We summarize the results and analyze the reasons, respectively.

As depicted in Figure 5, (i) generally, the location privacy guarantee increases with the decrease of privacy budget ϵ. (ii) UDPT performs better than its competitors. The reasons for the experimental results are as follows.

(i) According to the definition of differential privacy, a smaller privacy budget usually brings about more noise, a.k.a, larger randomness, being injected into the mobility features of the obfuscated trajectory dataset D_O. Consider the worst case where the privacy budget was the smallest. The trajectories in D_O would be highly random, and the mobility features in D_O would converge to population averages rather than maintaining the data utility in the actual trajectory dataset D_A. However, recall the description of the privacy metric in Section 5.4, the adversary was assumed to have already obtained the mobility features of population average as a prior belief before observing D_O. Therefore, in this case, the adversary gained no extra knowledge after observing D_O, which indicated that the strongest guarantee of privacy was provided. On the contrary, if the privacy budget were largest, the adversary would gain much knowledge after observing D_O, which indicated that the privacy guarantee was weakest. This explains the observation that the location privacy guarantee increases with the decrease of privacy budget ϵ.

(ii) AdaTrace generates obfuscated trajectories based on grid trajectories, where each obfuscated location would be uniformly chosen from the grid at random. Since the grid is a plane having infinite locations, the obfuscated locations have a great chance to be unique locations that only appear in D_O rather than D_A. Therefore, the posterior probability distribution $\Pr_{l|D_O}$ regarding the individuals who visit the sensitive location $l \in \mathcal{S}$ might have a larger difference from the prior probability distribution P_l. A larger difference implies a more serious threat of location privacy attacks, in other words, a lower privacy guarantee of AdaTrace. In contrast, UDPT's generation of obfuscated locations is guided under the multi-objective optimization algorithm as well as the Viterbi algorithm, which brings much less randomness to the obfuscated trajectory dataset than AdaTrace. This is the reason for the observation that UDPT performs better than its competitors in terms of privacy metrics.

6. Conclusions and Future Work

Releasing individual trajectories with high utility is a challenging task. On the one hand, lots of existing data analysis tasks heavily rely on both the geographical utility and semantic utility of the released trajectories. On the other hand, semantic utility could be used as side channels to conduct de-anonymization attacks and location inference attacks. Existing location privacy protection studies merely focus on the geographical utility and neglect the preservation of location semantics. As a result, not only did numerous data analysis tasks suffer poor performance, but individual identity and sensitive locations could also be disclosed. To remedy this problem, we propose a utility-preserving and differentially private mechanism for publishing trajectories (UDPT) with two novel features. First, it enables simultaneous preservation of both geographical and semantical utility by solving an optimization problem. Second, it provides a formal and provable privacy guarantee to thwart location inference attacks and de-anonymization attacks. To our best knowledge, it is the first work that ensures differential privacy and preserves both types of data utility. Extensive experiments in real-world datasets demonstrate UDPT's outperformance against two state-of-the-art competitors in terms of both data utility and privacy.

Our findings will promote the sharing of big trajectory data and improve the performance of the data analysis applications, e.g., semantic annotation and trajectory prediction, which heavily rely on trajectory data geographical utility as well as the semantic utility.

In the meantime, the findings will also shed more light on the study of location privacy protection.

In the future, we plan to extend our work from the following aspects. (i) The trade-off between geographical utility and semantic utility is an open problem. In this work, the data curators can express their preference for either type of utility by the weight parameter α in Equation (10) and then manually intervene the trade-off. Inspired by the automated model tuning and hyperparameter optimization in machine learning, in the future, we plan to develop an automated parameter tuning mechanism for finding an optimal trade-off between the two types of data utility. Similarly, the tuning of differential privacy budget is also an open problem, which is worth an in-depth study in the future. (ii) Recently, more cases of privacy disclosure, e.g., the Facebook–Cambridge Analytica data scandal in 2018, have suggested that data curators, such as location-based social networks, should take measures to protect individual privacy (https://en.wikipedia.org/wiki/Facebook%E2%80%93Cambridge_Analytica_data_scandal, accessed on 16 February 2022). A potential direction is to develop privacy tools for helping the curator to comply privacy protection regulation when releasing individual data. For example, Pereira et al. recently proposed a privacy tool for helping companies to comply with the general data protection regulation [36]. In the future, we plan to extend our work to a privacy-protecting trajectory data releasing middleware between the curators and the analyzers.

Author Contributions: Conceptualization, B.L.; methodology, B.L. and H.Z.; software, B.L. and H.Z.; validation, M.X.; formal analysis, B.L.; investigation, B.L. and H.Z.; resources, H.Z. and M.X.; data curation, H.Z. and M.X.; writing—original draft preparation, B.L.; writing—review and editing, H.Z. and M.X.; visualization, B.L.; supervision, H.Z. and M.X.; project administration, H.Z.; funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number 61772215) and the Wuhan Science and Technology Bureau (grant number 2018010401011274).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data analyzed in the work is publicly accessible.

Acknowledgments: The authors would like to acknowledge the efforts made by the editors and the reviewers, which greatly improved the quality of the paper.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

In this section, we provide detailed explanations of how we conducted our experiments for reproducibility, including location semantics acquirement, parameter setting, and implementations of the Laplace mechanism and the exponential mechanism.

Appendix A.1. Location Semantics Acquirement

In this work, a location is represented by a vector of GPS coordinates, e.g, (40.74224, −73.99386). The semantics of the location is represented by the Foursquare category tag corresponding to the location's GPS coordinates. Foursquare defines a hierarchical taxonomy of category tags. There are 10 parent category tags and hundreds of children in the taxonomy. In this work, we only use the 10 parent category tags, as shown in Table A1. Moreover, Foursquare provides developers with an application programming interface (API) named "Place Search" [27]. Given the GPS coordinates of a location, the API can returns the parent category tag of the location. Note that the developers need to apply a Foursquare account as well as an authentication key before calling the Foursquare APIs. There could be charges and rate limits.

Table A1. Foursquare category tags.

Arts and entertainment
Business and professional services
Community and government
Dining and drinking
Event
Health and medicine
Landmarks and outdoors
Retail
Sports and recreation
Travel and transportation

Appendix A.2. Parameter Settings

Parameters in our experiments are listed in Table A2. (i) In Algorithm 1, the (total) privacy budget of UDPT takes values in the interval $[0.1, 10]$, which is a widely-accepted convention in the literature. The division of the (total) budget is $1/3 : 1/2 : 1/6$ for Algorithm 2, 3 and 4, respectively. Note that a larger privacy budget (piece) usually brings about better preservation of data utility. We allocate larger budget piece to Algorithms 2 and 3, because their iterations consume more budgets for noise injecting. In particular, we allocate the largest budget piece to Algorithm 3 for simultaneously preserving the geographical and semantic utility as much as possible. (ii) In Algorithm 2, we set the number of cluster as 100. In this case, the diameter of a single cluster is around one to two kilometers, which can represent a real-world point-of-interest, e.g., a hospital. The maximum number of iterations in clustering, p, depends on the experimental datasets. In this work, $[20, 50]$ is a suitable interval for p. (iii) In Algorithm 3, we set the number of candidate obfuscated locations, M, as 6. A larger value results in better data utility while larger time cost. In our experiments, $M = 6$ can achieve an acceptable trade-off between the time cost and data utility. The previous work [29] related to genetic algorithm has demonstrated that without domain-specific information, the setting of $m = 200$, $m' = 10$, and $r \in [20, 40]$ generally leads to good results. We follow this setting in our experiments. The weight of geographical utility, α, indicates the data curator's preference for the geographical utility compared with the semantic utility; thus, it actually should be determined by the data curator. In this work, to demonstrate that UDPT's outperformance, in terms of semantic utility, we tune α until UDPT achieves similar geographical utility with its competitors. In this case, α is around 0.5. (iv) In Algorithm 4, the number of candidate obfuscated trajectories is set as 10. Since UDPT synthesizes N obfuscated trajectories for each actual trajectory in D_A, the data curator can exploit N to control the size of D_O. In this work, 10 is a moderate value for N.

Table A2. Parameters in the Experiments.

Symbol	Value	Meaning
ϵ	$[0.1, 10]$	(total) privacy budget of UDPT
ϵ_1	$\epsilon/3$	privacy budget piece allocated to Algorithm 2
ϵ_2	$\epsilon/2$	privacy budget piece allocated to Algorithm 3
ϵ_3	$\epsilon/6$	privacy budget piece allocated to Algorithm 4
K	100	number of clusters
p	[20, 50]	maximum number of iterations in clustering
M	6	number of candidate obfuscated locations
m	200	number of intermediate sets
m'	10	number of selected sets
r	[20, 40]	maximum number of iterations in Algorithm 3
α	0.5	weight of geographical utility in Equation (10)
N	10	number of candidate obfuscated trajectories

Appendix A.3. Implementation of Laplace Mechanism and Exponential Mechanism

Recall the Theorem 1, the Laplace mechanism is essentially a continuous random variable following the Laplace distribution with the location parameter $\mu = 0$ and the scale parameter $b = \Delta f/\epsilon$. The sensitivity Δf is a property of the query, which relies on the specific query that we used. ϵ is the privacy budget defined by the data curator. When employing the Laplace mechanism to ensure a location privacy protection algorithm ϵ-differentially private, we first construct the Laplace distribution with $\mu = 0$ and $b = \Delta f/\epsilon$. Then, we randomly sample a real-value noise following the Laplace distribution and add it to the real query result. The above two steps can be easily implemented by the standards library of programming language, e.g., the method "numpy.random.laplace" in Python with the parameters $loc = 0.0$ and $scale = \Delta f/\epsilon$.

Similarly, recall the Theorem 2, the exponential mechanism is essentially a discrete random variable following the discretized exponential distribution, of which the the probability of any realization r of the random variable is proportional to $\exp(\frac{\epsilon q(D,r)}{2\Delta q})$. q is a score function defined by the location privacy protection algorithm and $q(D, r)$ represents the score of a categorical outcome r of the algorithm based on the dataset D. The sensitivity Δq is a property of the score function. When employing the exponential mechanism to ensure the location privacy protection algorithm ϵ-differentially private, we first construct a categorical probability distribution of all the possible outcomes by normalizing the possibilities $\exp(\frac{\epsilon q(D,r)}{2\Delta q})$ for all outcomes $Range(q)$. Then, we randomly sample one or more outcomes following the distribution. The above two steps can also be implemented by the standard library of programming language, e.g., the method "numpy.random.choice" in Python with the parameters $r = Range(q)$ and p the probability distribution.

References

1. Wang, D.; Miwa, T.; Morikawa, T. Big Trajectory Data Mining: A Survey of Methods, Applications, and Services. *Sensors* **2020**, *20*, 4571. [CrossRef] [PubMed]
2. Xie, C.; Yu, D.; Zheng, X.; Wang, Z.; Jiang, Z. Revealing spatiotemporal travel demand and community structure characteristics with taxi trip data: A case study of New York City. *PLoS ONE* **2021**, *16*, e259694. [CrossRef] [PubMed]
3. Li, B.; Zhu, H.; Xie, M. LISC: Location Inference Attack Enhanced by Spatial-Temporal-Social Correlations. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 19–23 August 2019; pp. 1083–1092. [CrossRef]

4. Huang, J.; Luo, Y.; Xu, M.; Hu, B.; Long, J. pShare: Privacy-Preserving Ride-Sharing System with Minimum-Detouring Route. *Appl. Sci.* **2022**, *12*, 842. [CrossRef]
5. Qureshi, K.N.; Shahzad, L.; Abdelmaboud, A.; Elfadil Eisa, T.A.; Alamri, B.; Javed, I.T.; Al-Dhaqm, A.; Crespi, N. A Blockchain-Based Efficient, Secure and Anonymous Conditional Privacy-Preserving and Authentication Scheme for the Internet of Vehicles. *Appl. Sci.* **2022**, *12*, 476. [CrossRef]
6. Gursoy, M.E. Utility-Aware Synthesis of Differentially Private and Attack-Resilient Location Traces. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18, Toronto, ON, Canada, 15–19 October 2018; ACM: New York, NY, USA, 2018; pp. 628–637. [CrossRef]
7. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In Proceedings of the Third Conference on Theory of Cryptography, TCC'06, New York, NY, USA, 4–7 March 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 265–284. [CrossRef]
8. Ye, M.; Shou, D.; Lee, W.C.; Yin, P.; Janowicz, K. On the Semantic Annotation of Places in Location-Based Social Networks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, San Diego, CA, USA, 21–24 August 2011; Association for Computing Machinery: New York, NY, USA, 2011; pp. 520–528. [CrossRef]
9. Zhang, D.; Lee, K.; Lee, I. Mining hierarchical semantic periodic patterns from GPS-collected spatial–temporal trajectories. *Expert Syst. Appl.* **2019**, *122*, 85–101. [CrossRef]
10. Li, W.; Liu, X.; Yan, C.; Ding, G.; Sun, Y.; Zhang, J. STS: Spatial–Temporal–Semantic Personalized Location Recommendation. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 538. [CrossRef]
11. Montjoye, Y.A.d.; Hidalgo, C.A.; Verleysen, M.; Blondel, V.D. Unique in the Crowd: The privacy bounds of human mobility. *Sci. Rep.* **2013**, *3*, 1376. [CrossRef] [PubMed]
12. Oya, S.; Troncoso, C.; Pérez-González, F. Back to the Drawing Board: Revisiting the Design of Optimal Location Privacy-preserving Mechanisms. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, Dallas, TX, USA, 30 October–3 November 2017; ACM: New York, NY, USA, 2017; pp. 1959–1972. [CrossRef]
13. Tian, C.; Xu, H.; Lu, T.; Jiang, R.; Kuang, Y. Semantic and Trade-Off Aware Location Privacy Protection in Road Networks Via Improved Multi-Objective Particle Swarm Optimization. *IEEE Access* **2021**, *9*, 54264–54275. [CrossRef]
14. Naini, F.M.; Unnikrishnan, J.; Thiran, P.; Vetterli, M. Where You Are Is Who You Are: User Identification by Matching Statistics. *IEEE Trans. Inf. Forensics Secur. (TIFS)* **2016**, *11*, 358–372. [CrossRef]
15. Drakonakis, K.; Ilia, P.; Ioannidis, S.; Polakis, J. Please Forget Where I Was Last Summer: The Privacy Risks of Public Location (Meta)Data. In Proceedings of the 26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, CA, USA, 24–27 February2019. [CrossRef]
16. Xu, X.; Chen, H.; Xie, L. A Location Privacy Preservation Method Based on Dummy Locations in Internet of Vehicles. *Appl. Sci.* **2021**, *11*, 4594. [CrossRef]
17. Schlegel, R.; Chow, C.Y.; Huang, Q.; Wong, D.S. Privacy-Preserving Location Sharing Services for Social Networks. *IEEE Trans. Serv. Comput.* **2017**, *10*, 811–825. [CrossRef]
18. Guan, Y.; Lu, R.; Zheng, Y.; Shao, J.; Wei, G. Toward Oblivious Location-Based k-Nearest Neighbor Query in Smart Cities. *IEEE Internet Things J.* **2021**, *8*, 14219–14231. [CrossRef]
19. He, X.; Cormode, G.; Machanavajjhala, A.; Procopiuc, C.M.; Srivastava, D. DPT: Differentially Private Trajectory Synthesis Using Hierarchical Reference Systems. *Proc. VLDB Endow.* **2015**, *8*, 1154–1165. [CrossRef]
20. Bindschaedler, V.; Shokri, R. Synthesizing Plausible Privacy-Preserving Location Traces. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 546–563. [CrossRef]
21. Tan, R.; Tao, Y.; Si, W.; Zhang, Y. Privacy preserving semantic trajectory data publishing for mobile location-based services. *Wirel. Netw.* **2020**, *26*, 5551–5560. [CrossRef]
22. Shaham, S.; Ding, M.; Liu, B.; Dang, S.; Lin, Z.; Li, J. Privacy Preserving Location Data Publishing: A Machine Learning Approach. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 3270–3283. [CrossRef]
23. Li, B.; Zhu, H.; Xie, M. Quantifying Location Privacy Risks Under Heterogeneous Correlations. *IEEE Access* **2021**, *9*, 23876–23893. [CrossRef]
24. McSherry, F.; Talwar, K. Mechanism Design via Differential Privacy. In Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), Providence, RI, USA, 21–23 October 2007; pp. 94–103. [CrossRef]
25. Liu, C.; Chakraborty, S.; Mittal, P. Dependence Makes You Vulnberable: Differential Privacy Under Dependent Tuples. In Proceedings of the 23th Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, CA, USA, 21–24 February 2016; Volume 16, pp. 1–15. [CrossRef]
26. Martins, J.A.; Ochôa, I.S.; Silva, L.A.; Mendes, A.S.; González, G.V.; De Paz Santana, J.; Leithardt, V.R.Q. PRIPRO: A Comparison of Classification Algorithms for Managing Receiving Notifications in Smart Environments. *Appl. Sci.* **2020**, *10*, 502. [CrossRef]
27. Foursquare Venue Categories. 2022 Available online: https://docs.foursquare.com/docs/categories (accessed on 21 January 2022).
28. Mitchell, M. *An Introduction to Genetic Algorithms*; MIT Press: Cambridge, MA, USA, 1998.
29. Zhang, J.; Xiao, X.; Yang, Y.; Zhang, Z.; Winslett, M. PrivGene: Differentially Private Model Fitting Using Genetic Algorithms. In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, SIGMOD '13, New York, NY, USA, 22–27 June 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 665–676. [CrossRef]

30. Zheng, Y.; Zhang, L.; Xie, X.; Ma, W.Y. Mining Interesting Locations and Travel Sequences from GPS Trajectories. In Proceedings of the 18th International Conference on World Wide Web, WWW '09, Madrid, Spain, 20–24 April 2009; Association for Computing Machinery: New York, NY, USA, 2009; pp. 791–800. [CrossRef]
31. Lafferty, J.D.; McCallum, A.; Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, Williamstown, MA, USA, 28 June–1 July 2001; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2001; pp. 282–289. [CrossRef]
32. Brown, D.G.; Golod, D. Decoding HMMs using the k best paths: Algorithms and applications. *BMC Bioinform.* **2010**, *11*, S28. [CrossRef] [PubMed]
33. Liu, Y.; Wei, W.; Sun, A.; Miao, C. Exploiting Geographical Neighborhood Characteristics for Location Recommendation. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14, Shanghai, China, 3–7 November 2014; ACM: New York, NY, USA, 2014; pp. 739–748. [CrossRef]
34. Cho, E.; Myers, S.A.; Leskovec, J. 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, KDD '11, San Diego, CA, USA, 21–24 August 2011; ACM: New York, NY, USA, 2011; pp. 1082–1090. [CrossRef]
35. Niu, B.; Li, Q.; Zhu, X.; Cao, G.; Li, H. Enhancing privacy through caching in location-based services. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 1017–1025. [CrossRef]
36. Pereira, F.P.; Crocker, P.; Valderi, V.L. PADRES: Tool for PrivAcy, Data REgulation and Security. *SoftwareX* **2022**, *17*, 100895. [CrossRef]