VTDP: Privately Sanitizing Fine-grained Vehicle Trajectory Data with Boosted Utility

Bingyu Liu, Shangyu Xie, Han Wang, Yuan Hong, *Senior Member, IEEE,* Xuegang Ban, *Member, IEEE,* and Meisam Mohammady

Abstract—With the rapidly growing deployment of intelligent transportation systems (ITS) and smart traffic applications, vehicle trajectory data are ubiquitously generated, e.g., from GPS navigation systems, mobile applications, and urban traffic cameras. Analyzing such fine-grained data would greatly benefit the development of ITS and smart cities, yet pose severe privacy risks due to the recorded drivers' visited locations, routes, and driving habits. Recently, some privacy enhancing techniques were proposed to sanitize such data. However, such schemes have some major limitations – they either lack formal privacy notions to quantify and bound the privacy risks, or result in very limited utility, e.g., only a sequence of locations or aggregated information can be released (without retaining the speeds, accelerations and the timestamps of vehicles). In this paper, we propose a novel framework to sanitize the fine-grained *vehicle trajectories with differential privacy* (VTDP), which provides rigorous privacy protection against adversaries who possess arbitrary background knowledge. Our VTDP technique involves three phases of differentially private sampling, which sequentially generate all the three categories of data (besides a pseudo identity for each vehicle) – *position*, *moving*, *timestamps*. It also includes a *vehicle trajectory interpolation* procedure to further improve the output utility with the properties of fine-grained vehicle trajectory data. We conducted experiments on real vehicle trajectory datasets to validate the performance of our approach.

Index Terms—Differential Privacy, Fine-grained Vehicle Trajectory Data, Utility, Intelligent Transportation Systems.

1 Introduction

With the rapidly growing deployment of intelligent transportation systems (ITS) and smart traffic applications, vehicle trajectory data are ubiquitously generated from GPS navigation systems, mobile applications (e.g., Uber), urban traffic cameras, roadside unit and connected vehicles to record temporal pattern of locations, speeds and accelerations for each vehicle in a fine-grained manner [1]. Such fine-grained time series data can be collected and analyzed to significantly promote the development of intelligent transportation systems, urban traffic optimization (e.g., optimizing the mobility of urban traffic, and learning the signalized phases of traffic lights [45]) and smart cities.

However, directly releasing or sharing such datasets for analyses would pose severe privacy concerns to vehicles and their drivers [5], [40]. Specifically, sequences of locations can reveal a driver's frequently visited positions (e.g., residence, hospital) and preferred routes. Other attributes (i.e., speed and acceleration) can reveal his/her driving habits. Although vehicle/driver identities (i.e., VIN number and driver license numbers) have been replaced with pseudo-IDs in such datasets, privacy risks have not been addressed as re-identification attacks can still be applied to the dataset with certain background knowledge. For instance, if an adversary knows that a driver has visited some locations at specific times, even a small part of known traces can make the individual's entire data vulnerable to re-identification. After re-identification, it will be readily to track the driver/vehicle over any time period, learning all visited locations (e.g., hospital, gas station, office and residence) as well as his/her driving habits.

For this reason, in the past decade, privacy concerns in some *similar datasets* have attracted significant interests [5], [7], [10], [27], [39]. The existing techniques can be classified into two different categories: (1) data sanitization techniques [7], [27], [39], and (2) virtual trip lines (VTLs) [4], [5]. In the former category, each of the data sanitization techniques defines a privacy notion and proposes an algorithm to anonymize individuals or obfuscate the location traces while satisfying the defined privacy notion (e.g., generalization, suppression, or differential privacy [7], [27], [39]). However, most of such privacy preserving techniques can only generate either *spatially aggregated* data (e.g., traffic statistics [8], [38]) or a sequence of locations *(by omitting the vehicle moving attributes, e.g., speed and acceleration, and even the timestamps*) [7], [27], [39]. Thus, the output utility would be constrained, and the privately released data (without indicators for traffic flow) may not function many urban traffic analyses for developing smart cities, which request the fine-grained data disclosure with moving/traffic information and timestamps [1], [16].

In the latter category, Hoh et al. [4] proposed the idea of virtual trip lines (VTLs) for protecting privacy, which are geographic markers that indicate where vehicles should provide location updates in their trajectories (*which are only a small subset of the trajectories around the signalized traffic intersections in general*). Ban and Gruteser [5] further showed that VTLs can be utilized to regulate location and

- *B. Liu, S. Xie, H. Wang and Y. Hong are with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL. Corresponding Author: Yuan Hong, E-mail: yuan.hong@iit.edu*
- *X. Ban is with the University of Washington, Seattle, WA.*
- *M. Mohammady is with Concordia University, Montreal, Canada.*

Manuscript received April 28, 2019; revised September 29, 2019; accepted December 5, 2019.

speed reports, such that the data needs for intersection modeling (e.g., signal performance measurement) can be satisfied while simultaneously protecting privacy. However, the VTLs have the following limitations. First, the privacy risks in the output data cannot be formally quantified and bounded (e.g., via a privacy notion). Second, the output data are collected based on specific areas, and cannot span over the entire vehicle trajectories. Thus, the utility of the output data might be limited to only a few applications.

In summary, we argue that most of the existing work either do not rely on a formal privacy notion (e.g., VTLs based techniques [4], [5]), or result in very limited utility (e.g., not fine-grained, without moving attributes and timestamps). To address all the above limitations, we propose a novel privacy preserving technique to sanitize fine-grained vehicle trajectories (*all the attributes*) with differential privacy [12], [31], which provides rigorous privacy guarantee in datasets against arbitrary background knowledge. It ensures that adding or removing all complete trajectory of each vehicle (*all the attributes*) does not result in significant privacy risks.

1.1 Contributions

Our differentially private scheme randomly samples the output data without aggregation while satisfying the defined rigorous privacy notion. Therefore, the major contributions of this paper are summarized as follows.

- To the best of our knowledge, this is the first work that sanitizes fine-grained vehicle trajectory data under differential privacy guarantee to generate *vehicle pseudo IDs, coordinates, speeds, accelerations*, and *timestamps*. We note that our technique can output one *record-per-0.1 second* fine-grained trajectories for vehicles, the existing work on trajectory sanitization [7], [27], [39] or privacy preserving traffic flows [24], [33] mainly consider incomplete or coarse-grained data., e.g., counts and occupancy times measured by the installed loop detectors on highways.
- We propose a novel sanitization framework (namely, VTDP) that includes three phases to sample all the attributes in sequence with differential privacy. Our framework also *interpolates* data to further improve the output utility by any untrusted data recipient.
- Our VTDP framework is proposed based on sampling mechanisms, which satisfy *non-interactive* differential privacy [6], [18], [19], [25], [28]. Then, the non-aggregated output data (from non-interactive mechanisms) can be utilized for any utility-driven vehicle trajectory analysis such as traffic light signal phase learning [9] and queue length estimation [17]. Furthermore, we propose a novel multi-phase sampling scheme which can efficiently compute the output trajectories from our fine-grained data, which ensuring both privacy and utility (see Example 1).
- We conduct experiments on real world vehicle trajectory datasets [21] (e.g., the NGSIM data [1]) and validate the effectiveness of our scheme.

1. https://data.transportation.gov/api/views/8ect-6jqj

The remainder of this paper is organized as follows. Section 2 presents some preliminaries for our scheme. Section 3-5 illustrate the three phases of our differentially private algorithm, and give the privacy analysis. Section 6 discusses the data interpolation for boosted utility and the composition of differential privacy. We demonstrate the experimental results in Section 7, discuss the related work in Section 8, and conclude the paper in Section 9.

2 PRELIMINARIES

2.1 Fine-grained Vehicle Trajectory Data

TABLE 1
Fine-grained vehicle trajectory data

V-ID	Position			Moving		Time	
	ℓ	x	y	v	a	d	t
10	1	-72.2	1181.4	0	0	1	0.8
10	1	-73.38	1181.3	0	0	1	0.9
10	1	-75.54	1181.2	0	0	1	1.0
1001	2	4.1	163.7	18.4	1.2	1	57.8
1001	2	6.3	163.7	21.6	1.4	1	57.9
1001	2	9.9	164.7	15.1	-2.3	1	58.1
1005	1	-5.1	130.3	22.2	-1.9	1	60.2
	...	...	...	...	...	...	...

Table 1 presents an example of the fine-grained vehicle trajectory data, e.g., the NGSIM data collected from traffic cameras.[1] Such datasets include vehicle IDs (pseudo identities), lane ID ℓ, lateral/longitudinal coordinates of a position (x, y), speed v, acceleration a, day d, and time t, which belong to four different categories (*V-ID*, *position*, *moving* and *time*). Specifically, *position* consists of "lane" ℓ, "lateral/longitudinal coordinates" x and y where the coordinates x and y can uniquely determine its lane ℓ (thus we skip the lane in this paper). Moreover, we integrate "speed" (v) and "acceleration" (a) as the *moving* attributes, and the time includes "day" (d) and "time" (t). It is worth noting that all the position coordinates, speeds and accelerations are real numbers while the timestamps are discrete (e.g., with interval 0.1 second). To improve the output utility, all the position coordinates, speeds and accelerations can be approximated to discrete values (see Section 7.1).

Definition 2.1 (Vehicle Trajectory Data)**.** A collection of vehicles' fine-grained trajectories, each of which includes a pseudo-ID V_r denoting a vehicle, lane ID, coordinate (x, y), and moving speed v and acceleration a in day d at time t.

Vehicle trajectories can be formulated as above. In intelligent transportation systems (ITS), both vehicle speed and acceleration are considered as a part of vehicle trajectory data [1], [4], [5], [20]. Thus, we define "vehicle trajectory data" to differentiate it from the definition of "trajectories" in the existing trajectory sanitization works (related to location-based services) [7], [27], [39].

2.2 Privacy Notion

Before giving the definition of privacy notion, we first define vehicle trajectory in the dataset.

Definition 2.2. (VEHICLE TRAJECTORY) *Given a vehicle trajectory dataset D of n vehicles $V_1, \ldots, V_n$, vehicle trajectory $\Theta_r, r \in [1, n]$ is defined as all the tuples in D w.r.t. vehicle V_r.*

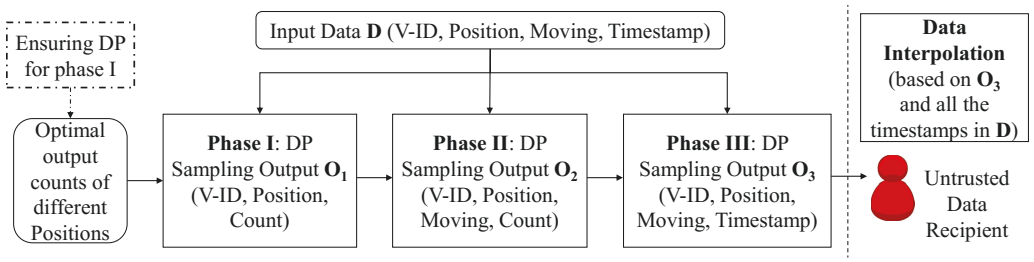


Fig. 1. Data sanitization framework for VTDP (after preprocessing)

With the definition of vehicle trajectory, we consider two datasets D and D' as neighboring inputs if they differ in one vehicle trajectory Θ_r, which is the complete traveling data corresponds to any vehicle V_r. Thus, our differential privacy definition [7], [21], [25] would provide the guarantee that adding or removing *any vehicle trajectory* does not result in significant risk to the privacy of dataset. Although two neighboring inputs D and D' differ in only one vehicle trajectory, the possible sets of outputs for applying a randomization algorithm $\mathcal{A}$ to D and D' might be different since the extra vehicle trajectory Θ_r may generate items in the output that cannot be derived from D or D' with $\mathcal{A}$ (e.g., the pseudo-ID of vehicle V_r, an extreme speed, or a unique timestamp in Θ_r). In this case, a relaxed differential privacy [15], [18] notion can be defined:

Definition 2.3 ((ϵ, δ)-differential privacy)**.** A randomization algorithm [23] $\mathcal{A}$ satisfies (ϵ, δ)-differential privacy if for all neighboring inputs D and D' and any set of possible outputs S, we have $Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} Pr[\mathcal{A}(D') \in S] + \delta$ and $Pr[\mathcal{A}(D') \in S] \leq e^{\epsilon} Pr[\mathcal{A}(D) \in S] + \delta$.

2.3 Sanitization Framework

We now present the framework for our vehicle trajectory data sanitization with differential privacy (VTDP). The overview of the framework is shown in Figure 1.

Multi-phase Sampling. Sampling the dataset is a way to achieve differential privacy. Accordingly, we propose a novel multi-phase sampling mechanism that randomly generates true values from the original input in three phases. Notice that, multi-phase sampling could improve the output utility in two folds: (1) generating complete attributes in the output (identical to the input) which can be used for any analysis, and (2) more tuples can be retained while satisfying the same differential privacy guarantee (as demonstrated in Example 1 and Figure 2).

- **Phase I**: Sampling the combinations of V-ID [2] and Position (V_r, P_i) from the *input data* D with the *specified output count for each position*. The optimal output counts of different positions will be derived (*maximizing the utility while satisfying the constraints of differential privacy guarantee for phase I*) to sample the V-IDs for each distinct position. Then, the output schema in phase I (denoted as O_1) is "V-ID, Position, Count". The details are given in Section 3.
- **Phase II**: Sampling the combinations of V-ID, Position and Moving (V_r, P_i, M_j) from the original *input data* D with the *phase I output* O_1. Then, the output schema in phase II (denoted as O_2) is "V-ID, Position, Moving, Count". The details are given in Section 4.
- **Phase III**: Sampling the original tuples (especially the timestamps) from the original *input data* D with the *phase II output* O_2. Then, the output schema in phase III (denoted as O_3) is "V-ID, Position, Moving, Timestamp". The details are given in Section 5.

Note that swapping the order of the three phases can also return a private output dataset but may result in reduced utility in vehicle trajectory data sanitization (*determined by the characteristics of the attributes*). For instance, if phase I samples V-IDs for timestamps, phase II samples positions based on the timestamps, and finally phase III samples moving values, the retained number of tuples would be less since many timestamps are associated with only a few vehicles during night time.

The following example based on Figure 2 further demonstrates the need for exerting the multi-phase sampling in scenarios when the dataset is fine-grained and diversified.

Example 1. *Figure 2 shows an excerpt of our vehicle trajectory dataset which goes under two different sampling mechanisms with the goal of achieving differential privacy. In the first mechanism, each record is independently sampled (all attributes together). As we use multinomial sampling in our scheme, all the records are going to be suppressed with a high probability (only those that have identical copies for all the attributes can be possibly retained).*

In the second mechanism, by breaking down the dataset into three sets of attributes (corresponding to three phases of sampling), the number of copies for each unique attribute significantly increases, e.g., as illustrated in the figure, the coordinates in four records $(-72.2, 1181.4)$, $(-73.38, 1181.3)$, $(-75.54, 1181.2)$, $(-76.5, 1181.2)$ *are very close (which can be approximated as the same position, as discussed in Section 7.1; close coordinates that are approximated as the same position are marked with the same color). This enhances the utility of the scheme through increasing the chance of retaining individual records. Accordingly, vehicle 10 can be picked with probability of* 0.75 *in every single sampling as it includes the same location (with very close coordinates) for three times and vehicle 1001 also includes it.*

Subsequent phases of sampling are then applied to estimate the values for the remaining attributes in a utility preserving manner.

Note that, the multi-phase sampling is expected to randomly generate a subset of the original dataset, in which

2. V-ID refers to vehicle pseudo identity in this paper.

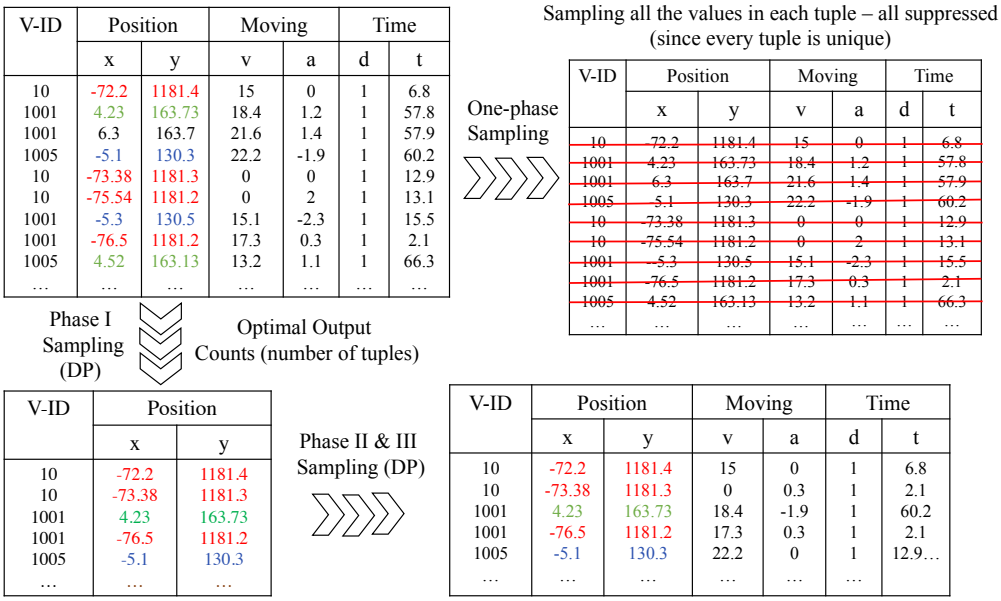


Fig. 2. An example of sampling fine-grained vehicle trajectory data (*one-phase* vs *multi-phase*).

each of the phases satisfies *differential privacy* (also detailed in the upcoming sections).

Boosting Utility. Our sanitization framework includes the following components to improve the output utility.

- **Multi-phase sampling** improves the utility (as described in Example 1).
- **Multinomial sampling** in phase I and II generates counts by preserving their original distribution (e.g., the distribution of vehicles visiting the same position, the distribution of moving values at the same position) in the output (see Section 3 and 4).
- **Utility** is maximized in phase I for multinomial sampling with differential privacy (see Section 3).
- **Trajectory interpolation**. Since each fine-grained trajectory posed by one vehicle has an equal-length interval between consecutive timestamps (e.g., 0.1 second), if any vehicle has sampled tuples in the output O_3, its *complete output vehicle trajectory* (at all the times given in the input data) can be approximately *interpolated* with the properties of vehicle trajectories (i.e., the formulas between speed, acceleration and times). This further improves the output utility, and can be conducted by any untrusted data recipient (without affect the privacy guarantee). The details are discussed in Section 6.

3 Phase I: Sampling (V-ID, Position)

VTDP in phase I, as shown in Figure 2, exerts sampling over the pair of vehicle IDs and their visited positions in the dataset. To preserve the distribution of vehicle IDs for each visited position, multinomial sampling is employed in phase I (as illustrated in Section 3.1). Next, as detailed in Section 3.2, we show that this notion of randomization can guarantee differential privacy with parameters ϵ and δ.

On the other hand, to boost the utility of the output, phase I in VTDP formulates a utility maximizing problem in which the optimal counts of specific positions emerge in the output, will be computed under the differential privacy constraints. Finally, we also ensure that no privacy violation from the optimization procedure occurs (see the discussion at the end of Section 3.3). Algorithm 1 presents the key steps for sampling phase I (denoted as $\mathcal{A}_1$), and Table 2 presents some frequently used notations in $\mathcal{A}_1$.

TABLE 2
Frequently used notations in phase I

V_r	the rth vehicle ID, $r \in [1, n]$
Ω	the set of distinct positions
Φ	the set of distinct moving values
Ψ	the set of distinct timestampes
$P_i \in \Omega$	the ith position, $\forall i \in [1, \lvert\Omega\rvert]$
$M_j \in \Phi$	the jth moving value, $j \in [1, \lvert\Phi\rvert]$
$T_k \in \Psi$	the kth timestamp, $k \in [1, \lvert\Psi\rvert]$
D, O_1	input data and output of phase I
$\lvert D\rvert, \lvert O_1\rvert$	the size of D and O_1
c_i	count of position P_i in the input
x_i	count of position P_i in the output
c_i^r	count of pair of (V_r, P_i)

3.1 Multinomial Sampling

Given any output count x_i for position P_i, multinomial sampling runs x_i *independent* trials to randomly pick V-IDs for P_i. Specifically, in every trial, a pair of V-ID and position (V_r, P_i) can be generated, and the probability for generating (V_r, P_i) is $\frac{c_i^r}{c_i}$ where the total count of P_i is referred to $c_i = \sum_{r=1}^{n} c_i^r$. After all the x_i trials, we denote the count of V_r in the output as x_i^r where $\sum_{r=1}^{n} x_i^r = x_i$. For example, in the input data D, V_1 has visited P_1 for 6 times, V_2 has visited P_1 for 2 times, V_3 has visited P_1 for 5 times, and V_4 has not visited P_1. Then, while sampling V-IDs for position P_1, in any trial, the probabilities for sampling V_1, V_2, V_3, V_4 are $\frac{6}{6+2+5+0}, \frac{2}{6+2+5+0}, \frac{5}{6+2+5+0}$ and 0. With the properties of multinomial sampling, the portion of the output counts for different pairs of V-ID and position (e.g., (V_1, P_1), (V_2, P_1),

Data: input D, privacy budgets ϵ, δ
Result: output O_1 as (V-ID, Position, Count)
1 **for** *vehicle* $r \leftarrow 1$ **to** n **do**
2 **for** *position* $i \leftarrow 1$ **to** $|\Omega|$ **do**
3 retrieve count c_i^r from D
4 **end**
5 **end**
6 **for** $r \leftarrow 1$ **to** n **do**
  `// DP for the Sampling`
7 derive constraints with the privacy budgets (ϵ, δ) for variables $\forall i \in [1, |\Omega|], x_i$ (the output count of all the vehicles for position P_i)
8 **end**
9 compute the optimal counts $\forall i \in [1, |\Omega|], x_i^*$ (while satisfying the constraints in Line 6-8)
10 **for** *position* $i \leftarrow 1$ **to** $|\Omega|$ **do**
11 randomly sample x_i^* V-IDs for position P_i using multinomial distribution: x_i^* independent trials (randomly picking a V-ID in each trial), where the probability of picking V_r in each trial is $\frac{c_i^r}{\sum_{r=1}^{n} c_i^r}$
12 **end**
  `//` (V_r, P_i) `is sampled for` x_i^r `times`
13 return the output O_1 as (V_r, P_i, x_i^r)

Algorithm 1: Sampling phase I $\mathcal{A}_1$

(V_3, P_1), and (V_4, P_1)) lies similar to those in the input data simply because $\forall r \in [1, |\Omega|]$, the expectation $E(x_i^r) = x_i \cdot \frac{c_i^r}{c_i}$ is proportional to $\frac{c_i^r}{c_i}$ (as the same x_i is applied).

3.2 Differential Privacy Guarantee

To investigate the differential privacy guarantee of multinomial sampling, we should explore the set of all possible outputs for any given input data D and any of its neighboring input data D' (differing in one vehicle V_r's vehicle trajectory Θ_r). As a result, we have two cases for neighboring inputs D and D': $D = D' \cup \Theta_r$ and $D' = D \cup \Theta_r$.

As discussed in Section 2.2, ϵ-differential privacy may not be achieved in the sampling since the probabilistic output O_1 may include item from D' yet cannot be from D (or vice-versa), e.g., V-ID V_r. Then, the relaxed notion (ϵ, δ)-differential privacy will be employed for phase I. Without loss of generality, we let $D = D' \cup \Theta_r$ (the other case $D' = D \cup \Theta_r$ will be discussed at the end of this subsection).

Now we divide the arbitrary output set $S \subseteq Range(\mathcal{A}_1)$ into S^+ and S^- where $V_r \in S^+$ and $V_r \notin S^-$ (note that S^+ *is formed with all the possible outputs with* V_r *while* S^- *does not include* V_r). Therefore, we can derive a sufficient condition for the randomization algorithm $\mathcal{A}_1$ and possible output O_1 (phase I) to bound $\frac{Pr[\mathcal{A}_1(D')\in S]}{Pr[\mathcal{A}_1(D)\in S]}$ and $\frac{Pr[\mathcal{A}_1(D)\in S]}{Pr[\mathcal{A}_1(D')\in S]}$ (Gotz et al. [15] also conducted similar study):

Theorem 1. *Given any neighboring inputs* D *and* D'*, if* $\forall O_1 \in S^-$*, inequality* $\frac{Pr[\mathcal{A}_1(D')=O_1]}{Pr[\mathcal{A}_1(D)=O_1]} \leq e^\epsilon$ *holds, then* $\frac{Pr[\mathcal{A}_1(D')\in S]}{Pr[\mathcal{A}_1(D)\in S]} \leq e^\epsilon$ *also holds.*

Proof. Since $\forall O_1 \in S^+$, $Pr[\mathcal{A}_1(D') = O_1] = 0$, we have

$$\begin{aligned} Pr[\mathcal{A}_1(D') \in S] &= \int_{\forall O_1 \in S^+} Pr[\mathcal{A}_1(D') = O_1] dO_1 \\ &\quad + \int_{\forall O_2 \in S^-} Pr[\mathcal{A}_1(D') = O_2] dO_2 \\ &\leq e^\epsilon \int_{\forall O_2 \in S^-} Pr[\mathcal{A}_1(D) = O_2] dO_2 \\ &= e^\epsilon Pr[\mathcal{A}_1(D) \in S^-] \\ &\leq e^\epsilon Pr[\mathcal{A}_1(D) \in S] \end{aligned}$$

This completes the proof. □

Similarly, we can prove that $Pr[\mathcal{A}_1(D) \in S] \leq \delta + e^\epsilon Pr[\mathcal{A}_1(D') \in S]$. This shows that we can ensure differential privacy by letting $\forall O_1$ in any S^- (viz. any output in $Range(\mathcal{A}_1)$ without V_r), $\frac{Pr[\mathcal{A}_1(D')=O_1]}{Pr[\mathcal{A}_1(D)=O_1]} \leq e^\epsilon$ in multinomial sampling, detailed as below.

3.2.1 Case (1): $\forall O_1 \in Range(\mathcal{A}_1)$ where $V_r \notin O_1$

Due to $V_r \notin O_1$, we have $Pr[A_1(D') = O_1] > 0$ and $Pr[A_1(D) = O_1] > 0$ (O_1 *can be generated from both* D *and* D' *with multinomial sampling*). At this time, we only need to sample V-IDs for the positions $\forall P_i \in D'$, which is a subset of D (otherwise, V_r might be included in O_1). Since sampling V-IDs for different positions is independent, we now examine two situations of all the positions in D'.

1) Position $P_i \in D' \setminus \Theta_r$. The probabilities for sampling any V-ID for P_i from D and D' are equal (since vehicle V_r's trajectory Θ_r does not include P_i). Thus,

$$\frac{Pr[(\mathcal{A}_1(D(P_i)) = O_1(P_i)]}{Pr[(\mathcal{A}_1(D'(P_i)) = O_1(P_i)]} = 1 \tag{1}$$

where $D(P_i)$ and $O_1(P_i)$ denote the position P_i's share of the input D and output O_1.

2) Position $P_i \in D' \cap \Theta_r$. At this time, sampling V-IDs for position P_i should ensure that the probability of picking V-ID V_r (out of all the vehicle IDs in D' and V_r) with multinomial sampling is bounded. Since picking V-IDs for x_i times trials is independent using multinomial distribution, we have following equations:

$$\begin{aligned} &\frac{Pr[\mathcal{A}_1(D'(P_i)) = O_1(P_i)]}{Pr[\mathcal{A}_1(D(P_i)) = O_1(P_i)]} \\ =&\frac{1}{(1 - \frac{c_i^r}{c_i})^{x_i}} = (\frac{c_i}{c_i - c_i^r})^{x_i} \end{aligned} \tag{2}$$

Given the output count $\forall i \in [1, |\Omega|], x_i$ for the ith position P_i, sampling vehicle IDs for each of the distinct position P_i is independent. For instance, while sampling for P_1, the sampling results can be "$(P_1, V_1, 5), (P_1, V_2, 3)$" where counts 5 and 3 are random. While sampling for P_2, the sampling results would be "$(P_2, V_1, 2), (P_2, V_2, 1)$" where counts 2 and 1 are random. Therefore, the privacy budget can be allocated for each sampling with sequential composition [29]. As a result, for each of the $|\Omega|$ different multinomial sampling (w.r.t. $|\Omega|$ different positions, respectively), the following constraint can be generated:

$$\max_{\forall \Theta_r \in D} \prod_{\forall P_i \in \Theta_r} (\frac{c_i}{c_i - c_i^r})^{x_i} \leq e^\epsilon \tag{3}$$

3.2.2 Case (2): $\forall O_1 \in Range(\mathcal{A}_1)$ where $V_r \in O_1$

In this case, the output O_1 would include item(s) from D but not D', e.g., $V_r \in \Theta_r$. Then, δ is defined to bound such probability $Pr[\mathcal{A}_1(D) \in S] \leq \delta$, which means

$$Pr[\mathcal{A}_1(D) \in S] \leq \delta \implies Pr[V_r \in \mathcal{A}_1(D)] \leq \delta \tag{4}$$

We now examine the probability $Pr[\mathcal{A}_1(D) \in S]$ while $Pr[\mathcal{A}_1(D') \in S]$ equals 0 (output includes V_r) as applying multinomial sampling to D and D', respectively.

$$\max_{\forall \Theta_r \in D} \prod_{\forall P_i \in \Theta_r} [1 - (\frac{c_i - c_i^r}{c_i})^{x_i}] \leq \delta \tag{5}$$

Per Definition 2.3, while specifying a small number δ, our algorithm ensures ϵ-differential privacy with high probability $(1 - \delta)$. Thus, we can simply remove the vehicle trajectories whose data results in a violation of Equation 5. Then, satisfcation of Equation 3 can ensure (ϵ, δ)-differential privacy for our Phase I sampling $\mathcal{A}_1$.

Discussion: in case of $D' = D \cup \Theta_r$, adding any vehicle trajectory Θ_r to input D to generate D'. Similarly, as long as the given $\forall i \in [1, |\Omega|], x_i$ satisfy Equation 3 (now c_i is derived from D' and c_i^r is the count of P_i in Θ_r), differential privacy is guaranteed.

3.3 Optimal Differentially Private Sampling

As analyzed in Section 3.2, if the output counts for all the positions $\forall i \in [1, |\Omega|], x_i$ satisfy inequality 3 (*inequality 5 will be employed in data preprocessing for small δ*), then the multinomial sampling to generate the output with schema *(V-ID, Position, Count)* would satisfy (ϵ, δ)-differential privacy.

Theorem 2. *Sampling in Algorithm 1 (Line 10-12) is (ϵ, δ)-differentially private if and only if inequalities 3 hold for all Θ_r.*

Proof. It is straightforward to prove that the probabilities that results in Case (2) for all the vehicles and positions are bounded by δ if inequality 5 holds (by setting δ in the preprocessing). In addition, as analyzed in Case (1), if inequality 3 holds for all Θ_r, per Theorem 1, we have:

$$e^{-\epsilon} \leq \frac{Pr[\mathcal{A}_1(D) \in S]}{Pr[\mathcal{A}_1(D') \in S]} \leq e^{\epsilon} \tag{6}$$

where S represents any set of possible outputs (without data from Θ_r). This completes the proof. □

Notice that, in a special case $c_i = c_i^r$ (the position in Θ_r is unique, and cannot be found in D'), x_i should be 0, otherwise, inequality 3 cannot hold.

Therefore, we should look for the output counts $\forall i \in [1, |\Omega|], x_i$ that satisfy Equation 3. Note that $\forall i \in [1, |\Omega|], x_i$ should be derived from D (or D') by subjecting to:

$$s.t. \begin{cases} \forall \Theta_r \in D, \prod_{\forall P_i \in \Theta_r} (\frac{c_i}{c_i - c_i^r})^{x_i} \leq e^{\epsilon} \\ \forall \Theta_r \in D, \prod_{\forall P_i \in \Theta_r} [1 - (\frac{c_i - c_i^r}{c_i})^{x_i}] \leq \delta \\ \forall x_i \geq 0 \text{ and } x_i \text{ is an integer} \end{cases} \tag{7}$$

While satisfying differential privacy, $\forall i \in [1, |\Omega|], x_i$ can have many possible results. We now seek for the optimal output counts for the differentially private sampling. A generic way of evaluating the utility is to measure the difference between the count distribution of all the positions in the input and output using distance metrics, e.g., ℓ_1-norm or ℓ_2-norm. However, the utility optimization based on such metrics may generate biased results towards the frequently visited positions (and the diversity of the positions may not be effectively preserved) [14].

To address such limitation, we define a universal utility measure (for multiple applications) for all the variables $\forall i \in [1, |\Omega|], x_i$ by following the KL-divergence [3] [18], [22], which evaluates the entropy-based distance between all the positions' distributions in the input data $(\frac{c_1}{|D|}, \frac{c_2}{|D|}, \ldots, \frac{c_{|\Omega|}}{|D|})$ and the output data $(\frac{x_1}{|O_1|}, \frac{x_2}{|O_1|}, \ldots, \frac{x_{|\Omega|}}{|O_1|})$ where $|D|$ and $|O_1|$ represent the total number of records in the input and output $|D| = \sum_{i=1}^{|\Omega|} c_i$ and $|O_1| = \sum_{i=1}^{|\Omega|} x_i$.

$$D_{KL} = \sum_{i=1}^{|\Omega|} \frac{c_i}{|D|} \big[\log(\frac{c_i}{|D|} \cdot \frac{|O_1| + |\Omega|}{x_i + 1}) \big] \tag{8}$$

Recall that minimizing the KL-divergence can maximally preserve the distribution/portion of each position in the output. Then, with multinomial sampling, the distribution/-portion of each combination of V-ID and position is expected to be preserved in the output as well. Since x_i may equal to 0 (in case of unique positions), the output counts in the KL-divergence are captured by approximating x_i with a close value $(x_i + 1)$ to avoid zero denominator.

Therefore, we can formulate an optimization problem to find the optimal multinomial sampling.

$$\begin{aligned} \min: \quad & \sum_{i=1}^{|\Omega|} \frac{c_i}{|D|} \big[\log(\frac{c_i}{|D|} \cdot \frac{|O_1| + |\Omega|}{x_i + 1}) \big] \\ s.t. & \begin{cases} \forall \Theta_r \in D, \prod_{\forall P_i \in \Theta_r} (\frac{c_i}{c_i - c_i^r})^{x_i} \leq e^{\epsilon} \\ \forall \Theta_r \in D, \prod_{\forall P_i \in \Theta_r} [1 - (\frac{c_i - c_i^r}{c_i})^{x_i}] \leq \delta \\ \forall x_i \geq 0 \text{ and } x_i \text{ is an integer} \end{cases} \end{aligned} \tag{9}$$

We can solve the above nonlinear programming (NLP) problem using pairwise linear approximation by converting the objective function to linear (the constraints can be simply converted to linear constraints) [18].

Differential Privacy for the Optimization. While applying Algorithm 1 to D and D' (solving the optimization problem 9) to get two sets of output counts $\forall i \in [1, |\Omega|], x_i^*$, and $\forall i \in [1, |\Omega|], x_i^{*'}$, respectively. In case that $D = D' \cup \Theta_r$, $\forall i \in [1, |\Omega|], x_i^*$ can ensure (ϵ, δ)-differential privacy for multinomial sampling. In case that $D' = D \cup \Theta_r$, $\forall i \in [1, |\Omega|], x_i^{*'}$ can ensure (ϵ, δ)-differential privacy. Apart from such privacy guarantee, we also need to make such two computed set of counts *indistinguishable*.

Specifically, we can consider the problem solving process as a query over the input data D or D', then the generic Laplace noise [2] $\frac{\Delta}{\epsilon'}$ can ensure ϵ'-differential privacy for the process of solving the problem [18], [19], where ϵ' is an additional privacy budget for this step, and sensitivity $\Delta = \max_{\forall D, D'} |x_i^* - x_i^{*'}|$ [13], [35]. Due to space limit, we skip

3. Optimizing the utility with KL-divergence can address the count bias as an entropy-based measure [14]. The optimization can preserve more distinct positions in the output as well as minimize the deviation between the distributions of all the positions in the input and output (ensuring that the data distribution in the output still lies close to that in the input). Notice that, KL-divergence is also used as the distance metric in case of similar scenarios. For instance, Acs et al. [3] measure the distance of the two probability distributions (count distribution in the input and output histograms).

the details of such generic mechanism here. In summary, we have the differential privacy guarantee for Algorithm 1.

Theorem 3. *Phase I is $(\epsilon + \epsilon', \delta)$-differentially private.*

Proof. This can be proven by the sequential composition [29] of solving the optimal problem and sampling. □

4 PHASE II: SAMPLING MOVING

In this section, we present the sampling phase II of our VTDP framework: randomly generating moving values by breaking down the counts for different pairs of V-IDs and positions to the counts for the triplets of V-IDs, positions and moving values. Furthermore, we study the differential privacy for phase II. Note that the required notations for sampling phase II are listed in Table 3.

TABLE 3
Frequently used notations in phase II

x_i^*	optimal count for Position P_i in phase I
x_i^r	sampled count of (V_r, P_i) in phase I
n'	number of vehicles in the output of phase I
γ^r	cardinality of sampled positions in phase I visited by V_r
$\theta_i(j)$	prior probability of sampling M_j for P_i (all vehicles)
θ_i	prior probability vector $\theta_i = (\theta_i(1), \ldots, \theta_i(\|\Phi\|))$
$\theta_i^r(j)$	posterior probability of sampling M_j for P_i and V_r
θ_i^r	posterior probability vector $\theta_i^r = (\theta_i^r(1), \ldots, \theta_i^r(\|\Phi\|))$
D_1, D_2	two datasets extract from the input D
$\lambda_i(j)$	count of (P_i, M_j) in D_1 (all the sampled vehicles)
$x_i^{r'}$	count of all the moving for (V_r, P_i) in D_2
$x_i^r(j)'$	count of (V_r, P_i, M_j) in D_2
O_2	output of phase II
$x_i^r(j)$	sampled count of (V_r, P_i, M_j) in phase II

4.1 Dirichlet-Multinomial Sampling

Similar to sampling phase I, the pair of visited position and moving values (i.e., speed and acceleration) for each vehicle in the trajectory data can be sampled with multinomial distribution which is expected to preserve the distribution of moving values associated with each position. More specifically, in phase II, for each vehicle, each moving data should be sampled from each of its visited positions (generated in phase I). Note that any count value for vehicle V_r and for position P_i is sampled as x_i^r in phase I, then x_i^r moving values (may include duplicates) will be sampled using an individual multinomial sampling in phase II.

Given n vehicles in the original input, after phase I, we denote the number of vehicles in the output as n' where $n' \leq n$ (since some V-IDs might not be randomly picked). Denoting the number of unique positions sampled for V_r in phase I as γ^r, there are $\sum_{r=1}^{n'} \gamma^r$ independent multinomial sampling in phase II, each of which is allocated for a unique pair of vehicle and one of its visited position. While sampling any moving values $M_j \in \Phi$ at position P_i for vehicle V_r, x_i^r independent trials will be tossed where the probabilities of possible sampling outcomes in every trial (denoted as *"probability vector"* $\theta_i^r = (\theta_i^r(1), \theta_i^r(2), \ldots, \theta_i^r(|\Phi|))$) will be learned from Dirichlet-Multinomial distribution [47] for the following reasons.

First, the distribution can integrate observations (drawn from the moving patterns posed by each vehicle in a particular position) and prior parameters (drawn from all the moving patterns at the same position). Therefore, considering the huge volume of moving patterns existed in the data, the posterior probability vector θ_i^r learned by the Dirichlet distribution would become significantly more accurate (e.g., in vehicle trajectory interpolation and analyses performed on the sanitized data). Second, sampling moving values with Dirichlet-Multinomial distribution does not result in *false moving values*. Specifically, if V_r has not visited P_i with moving value M_j, then the probability $\theta_i^r(j)$ would be 0 (since the corresponding observation is 0).

4.1.1 Probability Vector Learning

Before learning the probability vector, we extract two datasets from the input D (*which can minimize the privacy bound for phase II*, as illustrated in Section 4.2):

1) **Prior Data** D_1: a bipartite graph for every pair of position and moving (P_i, M_j) and the corresponding count $\lambda_i(j)$ – for deriving the prior distribution $\theta_i(j)$. The generation of D_1 includes two steps: (1) removing all the tuples inside each of the the unsampled trajectories (keeping only sampled data for n' vehicles), and (2) for every pair of position and moving (P_i, M_j), aggregating all the vehicles and timestamps' corresponding tuples to get count $d_i(j)$. Note that removing unsampled vehicles' data could ensure a tight privacy bound (e.g., $\epsilon = 0$) for phase II (as analyzed in Lemma 1).
2) **Observation Data** D_2: for each vehicle V_r, extracting its bipartite graph for each pair of its position and moving (P_i, M_j). Specifically, for each vehicle V_r, we extract sampled positions of V_r in phase I (γ^r distinct positions) and the corresponding tuples in D (tuples including unsampled positions will be removed), and aggregate all the timestamps for the corresponding tuples for (V_r, P_i, M_j) to get $x_i^r(j)'$.

Then, $\forall j$, $\lambda_i(j)$ and $\theta_i(j)$ can be derived from data D_1 while $\forall j, \theta_i^r(j)$ can be derived from data D_2. Per the Bayes rule, we can learn the posterior distribution for the probability vector: for each Vehicle V_r and its position P_i.

$$
\begin{aligned}
&Pr(\theta_i^r | M_1, \ldots, M_{|\Phi|}) \propto P(M_1, \ldots, M_{|\Phi|} | \theta_i^r) Pr(\theta_i^r) \\
&\propto \frac{\Gamma(\sum_{j=1}^{|\phi|} \lambda_i(j))}{\prod_{j=1}^{|\phi|} \Gamma(\lambda_i(j))} \prod_{j=1}^{|\phi|} (\theta_i^r(j))^{\lambda_i(j)-1} \frac{n!}{x_1^r(1)! \cdots x_i^r(|\phi|)!} \prod_{j=1}^{\phi} \theta_i^{x_i^r(j)'}(j) \\
&\propto \prod_{j=1}^{|\phi|} \theta_i^r(j)^{\lambda_i(j)-1+x_i^r(j)'}
\end{aligned}
$$

where constant $\frac{\prod_{j=1}^{|\phi|} \Gamma(\lambda_i(j))}{\Gamma(\sum_{j=1}^{|\phi|} \lambda_i(j))} = \frac{\Gamma(\lambda_i(1))\Gamma(\lambda_i(2)) \cdots \Gamma(\lambda_i(|\phi|))}{\Gamma(\lambda_i(1)+\lambda_i(2)+\cdots+\lambda_i(|\phi|))}$ and Gamma function $\Gamma(\lambda_i(j)) = (\lambda_i(j) - 1)!$. Notice that, the same prior probability vector θ_i^r is adopted for position P_i for all the vehicles, thus θ_i and θ_i^r are interchangeable. In addition, for V_r, the prior and posterior probabilities for most of moving values $M_1, \ldots, M_{|\Phi|}$ are 0 in practice. For simplicity of notations, we still use $M_1, \ldots, M_{|\Phi|}$ to represent the moving values.

4.1.2 Sampling Algorithm (Phase II)

We now present our sampling algorithm for phase II. First, the algorithm extracts D_1 and D_2 based on the output of phase I (O_1) and the original input D. Recall that,

1) D_1 is a bipartite graph with aggregated counts (in D) for every pair of (P_i, M_j) where the data of unsampled vehicles ($\forall V_r \in D \setminus D_2$) are not aggregated. Note that D_2 is the dataset including all the original tuples corresponding to the sampled output after phase I.
2) D_2 includes n' bipartite graphs (for n' sampled vehicles in O_1). Each vehicle's bipartite graph is extracted as its aggregated counts (in D) for every pair of (P_i, M_j) in O_1 (the output of phase I).

Second, the algorithm derives the prior probability vector of Dirichlet distribution and likelihood using D_1 and D_2. Thus, the posterior probability vector can be obtained using Bayes rule (using the expectation of the Dirichlet distribution [32]).

Finally, for each vehicle V_r and each of its visited position (e.g., P_i) in O_1, we apply multinomial sampling with its posterior probability vector and x_i^r trials. Algorithm 2 presents the details of sampling phase II.

Data: input D, phase I output O_1
Result: output O_2 as (V-ID, Position, Moving, Count)
extract D_1 and D_2 from D (using the Vehicles in O_1).
for $j \leftarrow 1$ **to** n **do**
  prior $P(\theta_i(j)) \leftarrow E[\theta_i(j)|\lambda_i(j)]$ where $E[\theta_i(j)|\lambda_i(j)][\theta_{ij}] = \frac{\lambda_i(j)}{\sum_{j=1}^{|\Phi|} \lambda_i(j)}$
  likelihood$\leftarrow \frac{x_i^{r'}(j)}{x_i^{r'}}$
end
Posterior $(\theta_i^r(j)) \leftarrow$ prior $(\theta_i(j)) \times$ likelihood $x_i^{r'}(j)/(x_i^{r'})$
foreach $V_r \in O_1$ **do**
  for $i \leftarrow 1$ **to** n **do**
    randomly sample x_i^r times moving values for vehicle V_r and position P_i using multinomial distribution: the probability of picking M_j in each trial is the posterior probability of $\theta_i^r(j)$
  end
end
return the output O_2 as $(V_r, P_i, M_j, x_i^r(j))$

Algorithm 2: Sampling phase II $\mathcal{A}_2$

4.2 Privacy Bound for Phase II

We now investigate the privacy bound for phase II, which samples x_i^r moving values for every pair of V-ID and position (V_r, P_i) where its count x_i^r is derived in phase I.

Lemma 1. *Phase II does not leak any additional information by sampling with the output of Phase I.*

Proof. We explore the privacy leakage by integrating phase I and II. Again, for two neighboring inputs D and D', w.l.o.g., we let $D = D' \cup \Theta_r$. In phase I, the probability of generating Case (2) (per Definition 2.3) is bounded by δ, which can be a negligible probability. Then, we only need to discuss Case (1) in phase I: $\forall O_1 \in Range(\mathcal{A}_1)$ where $V_r \notin O_1$, and investigate the privacy bound in phase II.

After phase I, the outputs (without V_r) derived from inputs D and D' are $(\epsilon + \epsilon')$-*indistinguishable*. Denoting the output for phase II as O_2, we first explore the multiplicative difference between probabilities $Pr[\mathcal{A}_2(D) = O_2]$ and $Pr[\mathcal{A}_2(D') = O_2]$. Specifically, as illustrated in Section 4.1, both D_1 and D_2 are extracted from D (or D' in the neighboring input case) in phase II (for learning the probability vector of multinomial sampling). In both D_1 and D_2, V-IDs is the baseline for extracting the tuples, whereas in the output of phase I: O_1, the position is the baseline. Since O_1 is indistinguishable for both inputs D and D' (both without V_r), each of two datasets D_1 and D_2 makes no difference in case of both D and D' (though D differs from D' in any vehicle trajectory Θ_r in phase I and II). Then, the probability vector would be indistinguishable for D and D', and thus we have:

$$\forall O_2 \in Range(\mathcal{A}_2), \frac{Pr[\mathcal{A}_2(D) = O_2]}{Pr[\mathcal{A}_2(D') = O_2]} = 1 \qquad (10)$$

Note that even if a new vehicle trajectory Θ_r' is added to D at the beginning of phase II to form D', data in Θ_r' will be suppressed while generating D_1 and D_2 for sampling (due to O_1). In this case, Equation 10 still holds. Similar to Theorem 1, given any possible output set S in phase II, we have $\frac{Pr[\mathcal{A}_2(D') \in S]}{Pr[\mathcal{A}_2(D) \in S]} = 1$.

Therefore, Phase II ensures 0-indistinguishability to randomly generate the output O_2. □

5 Phase III: Sampling Timestamps

In this section, we discuss how to sample the timestamps based on phase II output O_2, which includes the output count $x_i^r(j)$ for each pairs of position and moving (P_i, M_j) for V_r. Then, the timestamps sampling for the triplet (V_r, P_i, M_j) in phase III will be based on count $x_i^r(j)$. Indeed, phase III is not the same as the previous two phases, due to the uniqueness of timestamps. Specifically, for each timestamp, there exists exactly only one vehicle at the same position (which has been validated in our experimental data). On the contrary, one vehicle may visit the same location every day or stay at the one position over a period, thus the triplet of (V_r, P_i, M_j) may have multiple unique timestamps $T_k \in \{T_1, T_2, ...T_{|\Psi|}\}$ in D (denoting such count as $c_i^r(j)$). We then present our algorithm $\mathcal{A}_3$ by considering the above facts. Similar to phase II $\mathcal{A}_2$, phase III also extracts a dataset D_3 from D based on O_2:

- For all vehicles $\forall V_r \in O_2$, extract trajectories Θ_r from D to generate D_3.

For each triplet (V_r, P_i, M_j), the algorithm in phase III randomly picks $x_i^r(j)$ timestamps out of $c_i^r(j)$ unique timestamps from D_3 (*note that $c_i^r(j)$ are identical in D and D_3*). However, since $x_i^r(j)$ was randomly generated with multinomial sampling in phase I and II, $x_i^r(j)$ may exceed $c_i^r(j)$, though the probability of generating such extreme case is fairly low. Thus, we have to handle such extreme case in our algorithm $\mathcal{A}_3$ in the following two situations.

- If $x_i^r(j) \leq c_i^r(j)$, the algorithm simply picks $x_i^r(j)$ timestamps out of $c_i^r(j)$ unique timestamps from D_3.
- If $x_i^r(j) > c_i^r(j)$. The algorithm first picks all $c_i^r(j)$ timestamps out of $c_i^r(j)$ unique timestamps from D, and then randomly picks $x_i^r(j) - c_i^r(j)$ timestamps from other tuples which include position P_i and moving M_j (other vehicles). Note that the associated V-IDs for the latter picked timestamps will not be V_r. This ensures that all the tuples randomly selected from D are true tuples.

Data: input D, phase II output O_2
Result: output O_3 as (V-ID, Position, Moving, Timestamp)
extract D_3 from D (using the Vehicles in O_2).
foreach $V_r \in O_2$ **do**
  for $i \leftarrow 1$ **to** n **do**
    if $x_i^r(j) \leq c_i^r(j)$ **then**
      randomly pick $x_i^r(j)$ unique timestamps from $c_i^r(j)$ in D_3
    else
      randomly pick $x_i^r(j)$ unique timestamps from $c_i^r(j)$ in D_3
      randomly picks $x_i^r(j) - c_i^r(j)$ timestamps from other tuples in D_3 which include position P_i and moving M_j (other vehicles)
    end
  end
end
return the output O_3 as (V_r, P_i, M_j, T_k)

Algorithm 3: Sampling phase III $\mathcal{A}_3$

5.1 Privacy Bound for Phase III

Similar to phase II, phase III also ensures indistinguishability for any neighboring inputs D and D'.

Lemma 2. *Phase III does not leak any additional information by sampling with the output of Phase II.*

Proof. Given two inputs data D and D' where $D = D' \cup \Theta_r$ (or $D' = D \cup \Theta_r$), similar to D_1 and D_2 in phase II, the datasets (denoted as D_3) extracted from D and D' for sampling are indistinguishable, since O_2 is indistinguishable for D and D' after phase I and II, and data in Θ_r is suppressed in D_3 in any case. Then, the probabilities of randomly picking any timestamp (tuple) from the D_3 of D and D', and the count $x_i^r(j)$ are indistinguishable for any neighboring inputs D and D'. Thus, we have:

$$\forall O_3 \in Range(\mathcal{A}_3), \frac{Pr[\mathcal{A}_3(D) = O_3]}{Pr[\mathcal{A}_3(D') = O_3]} = 1 \qquad (11)$$

Similar to Theorem 1 and Lemma 1, given any possible output set S in phase III, we have $\frac{Pr[\mathcal{A}_3(D') \in S]}{Pr[\mathcal{A}_3(D) \in S]} = 1$. Therefore, phase III also ensures 0-indistinguishability to randomly generate the output O_3. □

6 Discussions

6.1 Vehicle Trajectory Interpolation

To further improve the output utility of our three-phase sampling, we propose a vehicle trajectory interpolation procedure in the VTDP framework to *approximately estimate the missing values at different times*.

As shown in Figure 1, the vehicle trajectory interpolation can be conducted by the untrusted data recipients (without affecting the privacy guarantee). Specifically, the interpolation is executed based on every two consecutive sampled tuples in trajectory θ_r (for the missing tuples between them). For instance, at time T_1 and T_6, two tuples are sampled in θ_r are sampled: "$\ell, (x_1, y_1), v_1, a_1, T_1$" and "$\ell', (x_6, y_6), v_6, a_6, T_6$". Then, all the tuples at T_2, T_3, T_4, T_5 can be interpolated using the two tuples at T_1 and T_6 (all the timestamps have equal intervals) with the following rules.

- The lane number of the first half of the tuples between T_1 and T_6 (viz. T_2 and T_3 in this example) is assigned as ℓ (same as T_1) while the second half (viz. T_4 and T_5) is assigned as ℓ' (same as T_6). If there are odd number of timestamps between two consecutive sampled tuples, the timestamp in the middle is considered as the first half.
- The position (x, y) for timestamps T_2, T_3, T_4, T_5 will be interpolated with equal distance between any two adjacent timestamps: $(x_2, y_2) = (x_1 + \frac{x_6 - x_1}{6-1}, y_1 + \frac{y_6 - y_1}{6-1})$, $(x_3, y_3) = (x_1 + \frac{2(x_6 - x_1)}{6-1}, y_1 + \frac{2(y_6 - y_1)}{6-1}), \ldots,$ $(x_5, y_5) = (x_1 + \frac{4(x_6 - x_1)}{6-1}, y_1 + \frac{4(y_6 - y_1)}{6-1})$.
- The interpolation for acceleration $a_2, \ldots, a_5$ follows the same way as position.
- Speed v for timestamps T_2, T_3, T_4, T_5 will be interpolated with the formula between speed, acceleration and moving time. Then, $v_2 = v_1 + a_1(T_2 - T_1)$, $v_3 = v_2 + a_2(T_3 - T_2), \ldots, v_5 = v_2 + a_2(T_3 - T_2)$.

It is worth noting that the above examples for vehicle trajectory interpolation are illustrated in case of driving in the same lane. If vehicles make turns or switch lanes, the missing values in the output data can also be interpolated in a similar manner.

Privacy Analysis. For any neighboring inputs D and D', since the probabilities of generating any O_3 from D and D' are bounded, adversaries (e.g., untrusted data recipient) cannot identify whether any vehicle trajectory Θ_r is included in the input or not – *indistinguishability*. Since such trajectory interpolation is a deterministic procedure (after receiving the output O_3) without any additional information, the adversaries cannot distinguish the interpolated outputs from D and D' either. Thus, the vehicle trajectory interpolation does not affect the differential privacy guarantee of our VTDP framework (and it can be performed by any untrusted data recipient).

6.2 Composition of Differential Privacy

Overall, the differential privacy for all the four major components of VTDP (computing optimal counts, sampling phase I, II and III) follows sequential composition [29]. We now discuss the composition and the privacy bounds step by step in our framework.

1) Computing the optimal counts (for sampling phase I): this step satisfies ϵ'-differential privacy.
2) Multinomial sampling to generate O_1 (sampling phase I). Sampling V-IDs for each position is independent but associates multiple positions with each V-ID. Thus, sampling phase I for each position follows sequential composition (as discussed in Section 3.2). This step satisfies (ϵ, δ)-differential privacy.
3) Dirichlet-Multinomial sampling to generate O_2 (sampling phase II). Sampling moving values for every pair of position and vehicle ID is independent (generating disjoint outputs), thus sampling phase II for every pair of position and vehicle ID follows parallel composition of differential privacy. This step has also been proven to satisfy 0-differential privacy (per Lemma 1).
4) Sampling timestamps to generate O_3 (sampling phase III). Similar to phase II, sampling timestamps for every pair of position and moving in θ_r also follows parallel composition of differential privacy. This step has also been proven to satisfy 0-differential privacy (per Lemma 2).

Theorem 4. *VTDP satisfies $(\epsilon + \epsilon', \delta)$-differential privacy.*

Proof. This can be proven by the sequential composition [29] of three sampling phases. □

6.3 Protection against Re-identification

We now discuss the re-identification attack to the sanitized dataset of VTDP. Assume that an adversary possesses arbitrary background knowledge on a specific vehicle V_r, e.g., knowing a large portion of places that the vehicle/driver has visited. While providing the differential privacy guarantee by VTDP, the probabilities of generating any output from D (with such vehicle's data) and D' (without such vehicle's data) are indistinguishable. Thus, the adversary cannot identify if such vehicle is included in the dataset from the output (since such output can also be obtained even if all the known places are not included in the input). At this time, knowing a large portion of places the vehicle/driver has visited cannot facilitate the re-identification.

6.4 Application to Sanitizing Other Datasets

Recall that phase I in our VTDP samples a probabilistic output with the attributes V-ID, position and count. Then, phase II samples the moving values to be associated with the V-ID and position. Finally, phase III samples the timestamps to be associated with the V-ID, position and moving values. The sanitization is not dependent on the number of fixed attributes. In other words, if more attributes are attached with the vehicle trajectories (e.g., distance to the traffic signal [45]), an output can be generated with the same number of tuples as the output of phase I. Following the above property of VTDP, we can apply our VTDP (via multi-phase sampling) to sanitize other datasets, such as generic microdata [28] and network data [34].

7 Experimental Results

7.1 Experimental Setup

Dataset. We conduct experiments on the NGSIM dataset [1], which is a real world fine-grained vehicle trajectory data with "lane, coordinates (x, y), speed, acceleration, day, time". The experimental dataset includes 1,809 distinct vehicles, each of them consists of 479,763 tuples in an arterial road (Peachtree Street in Atlanta, GA). The time interval for collecting data from each vehicle is 0.1 second. Table 4 presents the characteristics of our experimental dataset.

TABLE 4
Characteristics of the dataset

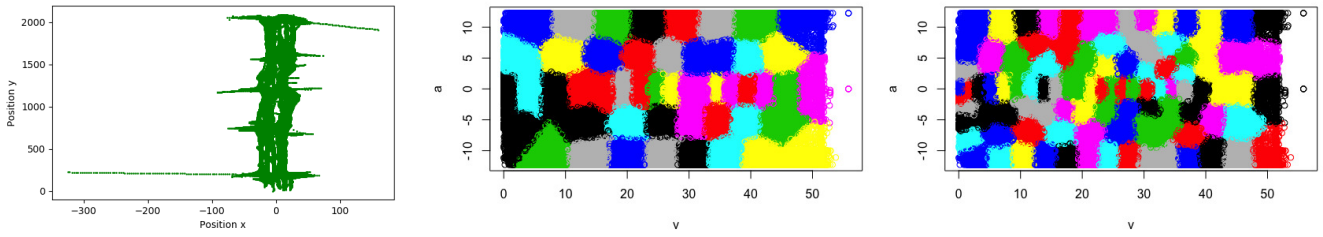
	Distinct #	Min	Max
Vehicles ID	1,809	n/a	n/a
Lane ID ℓ	7 (in multiple roads)	n/a	n/a
x (lateral)	66,336	-325.65	160.90
y (longitudinal)	372,003	0.0	2094.07
speed v	5,250	0.0	55.82
Acceleration a	2,451	-12.27	12.27
Day d	3	1	3
Time t	10,326	0.3	1,032.8

Data Approximation. Due to the fine-grained property of vehicle trajectories, two different values of any attribute might be extremely close, and can be approximated as the same value. For instance, since the distance between two coordinates (-72.2,1181.4) and (-73.38,1181.3) is very small, they can be approximated as the same location. Furthermore, moving values (20ft/s, 1.2ft/s^2) and (22ft/s, 1.1ft/s^2) may represent very similar moving attributes on the road. Therefore, we preprocess such fine-grained dataset by approximating close values in the raw data.

- First, all the *positions* (different combinations of ℓ, x, and y) can be partitioned with the equal size blocks (e.g., using the average length of vehicles), each of which can be approximated as a distinct position. All the coordinates falling into each block share the same position (e.g., the centroid coordinates). Then, we denote such positions as $P_1, P_2, \ldots, P_{|\Omega|} \in \Omega$ where Ω represents the universe of positions and $|\Omega|$ represents its cardinality.
- Second, we can also cluster all the *moving* values (different combinations of v and a) to approximate the moving status of vehicles (e.g., identify K different groups of moving status using K-means clustering [46]). All the combinations of speed and acceleration in the same cluster share the same moving data (e.g., the mean of the cluster). Then, all the distinct approximated moving values are denoted as $M_1, M_2, \ldots, M_{|\Phi|} \in \Phi$ where Φ represents the universe of the approximated combination of speed and acceleration, and $|\Phi|$ denotes its cardinality. Note that K can tune the granularity of the data in the approximation.
- Finally, the day and time are also fine-grained with equal length interval (e.g., 0.1 sec in the NGSIM data), then we consider them as the index of each vehicle's trajectories, and all the unique combinations of day and timestamp are denoted as $T_1, T_2, \ldots, T_{|\Psi|} \in \Psi$ where Ψ is the universe of day and time and $|\Psi|$ denotes its cardinality.

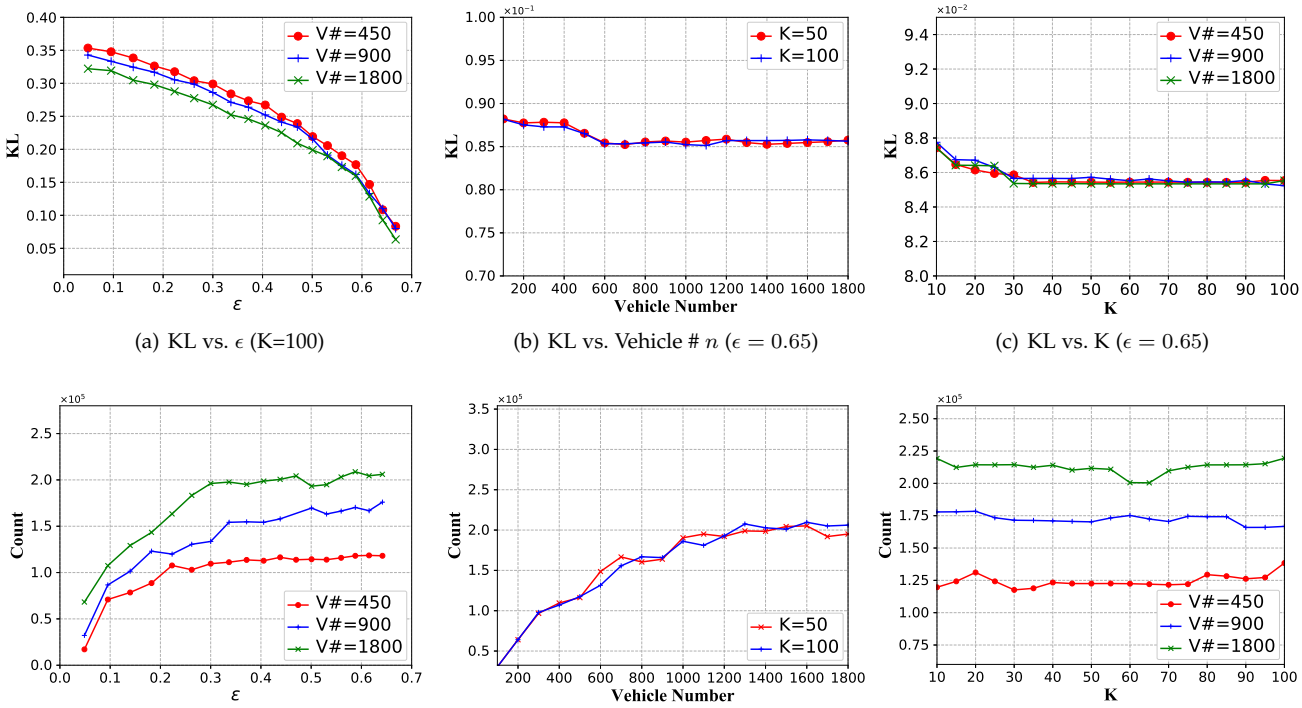
As a result, some representative positions are plotted in Figure 3(a) which demonstrates the traffic flow of the arterial road (note that many vehicles make turns at the intersections). For such fine-grained data, we approximate close values using clusters (described above). The coordinates of the positions are approximated by the equal size blocks (16.6ft×16.6ft) in coordinate axes. Since every pair of coordinates (x, y) can uniquely identify a position and the corresponding lane, we skip the lane in the plots. Furthermore, all the combinations of speed and acceleration are plotted in Figure 3(b) and 3(c) (clustered by K-Means where K=50 and 100 in the preprocessing while approximating each cluster as a distinct moving value) where the data points inside each cluster are marked with the same color.

Parameters. We evaluate the utility of our VTDP technique with different privacy bounds for $(\epsilon + \epsilon', \delta)$-differential privacy. We set $\epsilon \in [0.05, 0.65]$ and $\delta = 0.01$. In addition, since ϵ'-differential privacy for computation of optimal counts in phase I follows generic Laplace mechanism [13], we do not evaluate the utility on different ϵ'. Instead, we set $\epsilon' = \ln(2)$. We also test the output utility on different number of vehicles, and different K used in approximat-



(a) Coordinates of Positions (b) 50 Clusters for Speed and Acceleration (c) 100 Clusters for Speed and Acceleration

Fig. 3. Positions (x ft and y ft), speed (v ft/s) and acceleration (a ft/s^2) in the experimental data (Peachtree Street in Atlanta, GA)



(a) KL vs. ϵ (K=100) (b) KL vs. Vehicle # n ($\epsilon = 0.65$) (c) KL vs. K ($\epsilon = 0.65$)

(d) Output Count ($\times 10^5$) vs. ϵ (K=100) (e) Output Count ($\times 10^5$) vs.Vehicle # ($\epsilon = 0.65$) (f) Output Count ($\times 10^5$) vs. K ($\epsilon = 0.65$)

Fig. 4. Output utility vs. different parameters

ing speed and acceleration. Then, we set vehicle number $n \in [100, 200, 300, \ldots, 1800]$ and K $\in [10, 15, \ldots, 95, 100]$.

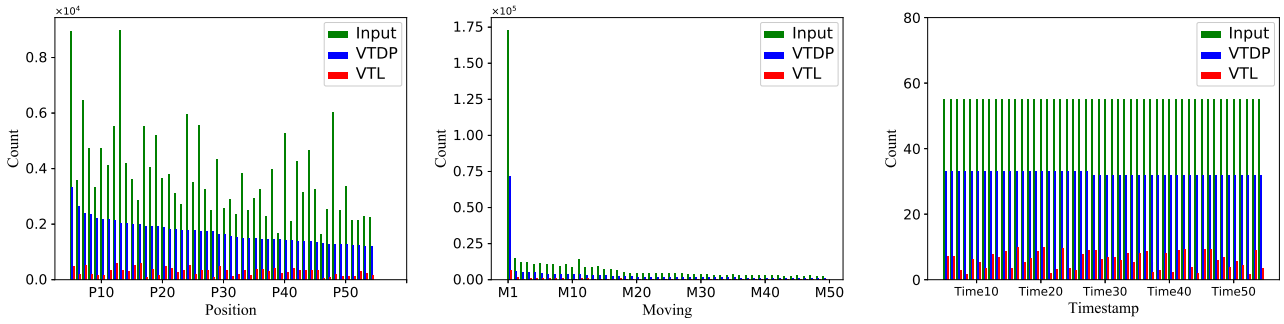
Platform. All the programs were implemented in Python 3.6.4 and tested on an HP PC with Inter Core i7-7700 CPU 3.60GHz and 32G RAM running Microsoft Windows 10 OS.

7.2 Utility Evaluation

We first evaluate the output utility using the KL-divergence measure and the total output counts (after interpolation). Notice that, in our VTDP framework, phase I determines the V-IDs and the total output count for each vehicle in O_1, O_2, O_3 while phase II and III sample other attributes by expanding the full tuples based on the output of phase I and the data distribution in the input. Therefore, the minimized KL-divergence in phase I (the objective function of the optimization problem) can be an effective measure for the overall output utility.

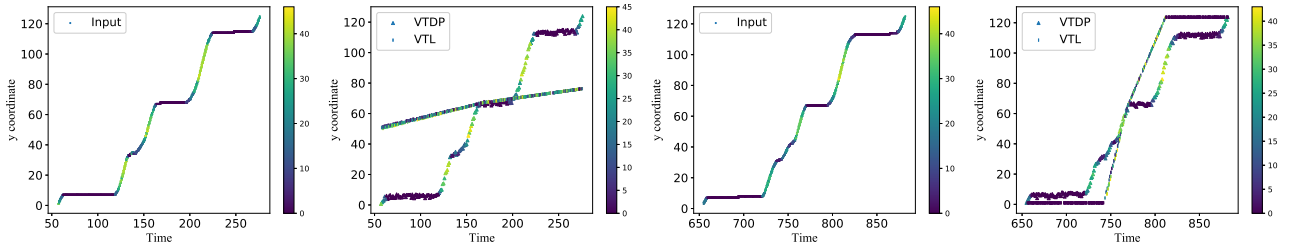
Figure 4(a) shows the KL-divergence results on varying privacy bound ϵ (given δ and ϵ') in case of different size of the input (different number of vehicle trajectories). As the number of vehicles n increases (from 450 to 1800), the utility performs better given the same privacy bound. However, as large privacy bounds are given, the KL-divergence results are quite close for different number of vehicles (as shown in Figure 4(a) and 4(b)). We observe that the KL-divergence for approximating the speed and acceleration is almost steady when K changes (see Figure 4(c)).

Our VTDP framework can generate a large number of output tuples via data interpolation where most of the interpolated tuples can be close to the original tuples (since the sampled tuples span over the entire trajectory for most of the vehicles). In the same group of experiments as Figure 4(a)-4(c), we plot the corresponding output counts in Figure 4(d)-4(f). The total count of tuples increases as the privacy bound ϵ, and/or number of vehicles n increases. The param-



(a) Position Counts (b) Moving Counts (c) Timestamp Counts

Fig. 5. Retained counts (top 50 frequent) in the output data (VTDP vs. VTL)



(a) Original (V-ID 95) (b) VTDP vs. VTL (V-ID 95) (c) Original (V-ID 1170) (d) VTDP vs. VTL (V-ID 1170)

Fig. 6. Output trajectory comparison (two representative vehicles)

eter K for approximating moving values does not affect the output counts. Note that the output utility slightly fluctuates since our VTDP is a multi-phase randomization framework (though the results have been averaged for 5 times). It is worth noting that the total output count is not very close to the input count in case of strong privacy guarantee (i.e., small ϵ for differential privacy). This may also occur in many other high-dimensional data sanitization (e.g., search queries [19], and trajectories [7]) and the output counts can be further enlarged by relaxing the privacy budget due to the tradeoff between privacy and utility. Indeed, since the data distributions of the input and output can be close after the sanitization, the output can still accurately function many applications, as illustrated in Section 7.3.

7.3 Trajectories Comparison

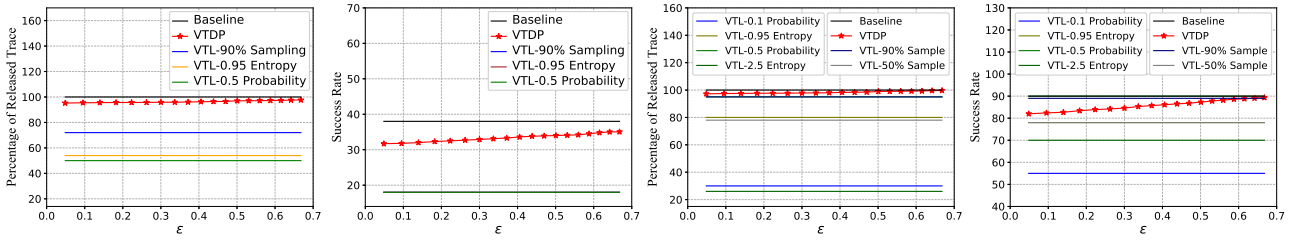
Besides quantitatively measuring the output utility, we also compare the utility of our VTDP technique with the existing privacy preserving approach ("VTL") [4], [5]. [4] We perform two groups of comparisons. First, in Figure 5(a)-5(c), we plot the top 50 frequent distinct positions (coordinates), moving values (approximated combinations of speed and acceleration), and timestamps (day and time) in one of our experimental results (ϵ=0.65, n=1800, K=100). The counts of such positions, moving values, and timestamps are well preserved after interpolation in our VTDP framework. Note that the interpolation is based on timestamps in the input (considering timestamps as the index of tuples), the counts of all the distinct timestamps are quite close, but slightly smaller (compared to VTL) than that in the input simply because some of the tuples for each vehicles have not been sampled.

Second, we apply the same interpolation to both VTDP and VTL (discussed in Section 6.1), and then we compare the results obtained for the output data posed by each specific vehicle. More specifically, in Figure 6, we plot a part of the trajectories of two representative vehicles (e.g., Vehicle 95 and 1170) in the arterial road. The y axes in Figure 6 show the longitudinal coordinates of the positions (note that lateral coordinates have negligible changes in the trajectories in that arterial road, we thus skip it for better visualization). The x axes in Figure 6 show the timestamps in sequence. The color bar presents the speed at different times. The results demonstrate that our VTDP technique can well preserve the trajectories and moving data (e.g., speed) – the trajectories for the two vehicles lie very close to the input compared to the interpolated results of VTL.

7.4 Comparison via Queue Length Estimation

We also evaluate that sanitized vehicle trajectories can still be effectively used for traffic modeling. Then, we apply real world traffic modeling applications, e.g., queue length estimation [5], [17] (which predicts the queue length at the traffic intersections) to our sanitized vehicle trajectories (generated by VTDP) and the output data generated from VTL techniques [4]. Thus, we compare the queue length estimation results derived from the VTDP output with the VTL outputs.

4. The utility of VTDP and other differentially private trajectory sanitization techniques (e.g., [7], [27], [39]) are incomparable, since the output of VTDP is generated with *more attributes* (and not aggregated).



(a) Released Trace (50%) (b) Success Rate (50%) (c) Released Trace (100%) (d) Success Rate (100%)

Fig. 7. Queue length estimation – penetration rate (50% or 100%): percent of vehicles in data collection (e.g., by mobile sensors, traffic cameras)

In literature, there are three different VTL techniques established based on different criteria (e.g., sampling, entropy, and probability) [4]. First, *sampling based VTL technique* randomly captures a portion of the traces (say 50%) at each VTL zone. Second, *probability based VTL technique* treats tracking probability as a privacy metric to generate the VTLs. It ensures that the released traces should have low tracking probability, e.g., 0.2 probability indicates that no more than one out of five vehicles can be successfully tracked. Third, *entropy based VTL technique* calculates the entropy value for a specific location trace for all possible vehicles that previously passed for specific VTL zones. Higher level of entropy gives higher confusion and better privacy.

In Figure 7, we demonstrate the queue length estimation results (VTDP vs. different VTLs) with measures *percentage of released trace* and *Success Rate*. The success rate of queue length estimation (also adopted in [5]) indicates the performance of traffic modeling application, which defines as the percentage of cycles that the proposed algorithms can be successfully applied (i.e., cycles that have 2 or more samples of queued vehicles) [5]. Specifically, the "Baseline" results are captured with all the data around the VTL zones. While testing our sanitized data using queue length estimation application with different ratios of vehicles involved in the data collection (50% and 100% penetration rate), the results are very close to the baseline (as shown in Figure 7). Furthermore, compared to three different VTL techniques with different parameters (e.g., 90% sample, 50% sample, 0.5 probability, 0.1 probability, 2.5 entropy, 0.95 entropy), the vehicle trajectories generated by VTDP can provide better success rates for queue length estimation at signalized intersections. We can also observe that both the percentage of released trace and success rate lie closer to the baseline as ϵ increases (better utility with increased privacy budget).

Overall, our VTDP technique can generate vehicle trajectories with better utility than the state-of-the-art while ensuring stronger privacy guarantee. Recall that, applying some existing techniques (e.g., [7], [27], [39]) to fine-grained vehicle trajectories generates either incomplete attributes (suppressing moving values and timestamps) or aggregated data (e.g., for locations/positions). Thus, the results are incomparable with our VTDP technique.

7.5 Computational Costs

Since our VTDP algorithm has $O(n^2)$ complexity: $O(n^2)$ for optimal counts computation, $O(n)$ for three phases of sampling, and $O(n)$ for data interpolation, the vehicle trajectory data can be sanitized with high efficiency and scalability. Thus, we do not present such low computation costs due to space limit.

8 Related Work

Vehicle trajectory data generated in mobile apps, traffic monitoring cameras, and GPS navigation system have great values to function intelligent transportation systems and smart cities. However, the privacy concerns in such data have received much attention, and have never been adequately addressed. In the prior work, some privacy techniques (including data sanitization [7], [27], [37], [39] and VTLs [4], [5]) are proposed to moderate the privacy issues. However, VTL techniques cannot fully protect the privacy (with provable guarantee) and existing data sanitization techniques cannot generate satisfactory fine-grained vehicle trajectory data for urban traffic modeling [16]. To address such limitations, our proposed VTDP technique satisfies the differential privacy with boosted utility.

Dwork et al. [12], [13] first proposed the rigorous privacy definition of differential privacy, which is a randomization algorithm which guarantees that for any two neighboring input datasets, the probabilities of generating any output from two inputs are bounded. This notion provides sufficient privacy protection for users regardless of the prior knowledge possessed by the adversaries. In the past decade, this has been extended to data release in different contexts. For instance, McSherry et al. [30] solved the problem of producing recommendations from collective user behavior while providing differential privacy for users. Wang et al. [43], [44] proposed a differentially private schemes for video analytics. In particular, some non-interactive differentially private data sanitization techniques [6], [19], [28] are very relevant to our work. Li et al. [28] identified the weakness of k-anonymity and proposed a privacy notion of safe k-anonymization to address such vulnerability by applying random sampling to meet k-anonymity and differential privacy. Bild et al. [6] proposed an approach for implementing the traditional data anonymization algorithm (k-anonymity) with differentially private components where k-anonymization was employed in order to reduce the added noise. Both techniques generate sanitized outputs for generic datasets while satisfying k-anonymity and differential privacy simultaneously. In addition, Hong et al. [19] proposed a multinomial sampling based approach to generate sanitized search logs while maximizing the output utility. Phase I in our VTDP framework is inspired from

such work where trajectory data (e.g., position coordinates) are substantially different from the search logs. Also, phase II and III (in VTDP) sample additional values (e.g., speed, acceleration and timestamps) based on the output of the multinomial sampling and the original input (whereas the timestamps in [19] are not published). Also, vehicle trajectory data provides properties to further improve the output utility via data interpolation.

Furthermore, previous work on preserving privacy in practical transportation systems is sparse. Hoh et al. [17] rely on a notion of privacy, k-anonymity, that is not particularly strong at preserving location privacy [42]. In particular, they focus on privacy for individual measurements, and thus do not directly offer formal protection for users transmitting time series such as location traces. Some research on privacy for location-based services, e.g., [38], can be considered somewhat related to our work. These works are typically concerned with perturbing GPS location traces to provide privacy while reconstructing some aggregate statistics, e.g, average density. However, they generally either do not rely on a formal definition of privacy, or consider simply the minimization of mutual information between the users' private data and the published data, which ignores the crucial issue of side information. In addition, Li et al. [26] has quantified the privacy leakage while sharing the locations in mobile social networks, and proposed a system-level solution (i.e., SmartMask) to prevent the location privacy breaches. Similar to our work, Ny et al. [33], [36] consider more traditional static sensors, e.g., single loop detectors. However, such techniques do not collect the fine-grained trajectory data, and fine-grained vehicle trajectories are not generated for output, either.

In intelligent transportation systems, privacy preserving VANET (Vehicular Ad-hoc Networks) applications [11], [41] may generate similar datasets. However, our VTDP significantly differs from such works. Specifically, our VTDP focuses on the differentially private vehicle trajectory (including speed and acceleration) data sanitization. In such case, a data curator applies the proposed offline algorithm to generate a publishable dataset, which can be shared to any untrusted party. However, VANET focuses more on real-time communications between vehicles and/or infrastructure in a short range (e.g., real time computation/communication for road safety, and navigation) where privacy is generally ensured by cryptographic schemes [41].

9 Conclusion

As the rapidly growing deployment of intelligent transportation systems (ITS) and smart traffic applications, fine-grained vehicle trajectory datasets are generated from everywhere in our real life, e.g., GPS navigation systems, mobile applications, and urban traffic cameras. Although these data are extremely valuable for the ITS development, privacy risks also arise if such data are not properly sanitized before release for analysis. Recently, some researchers have proposed techniques to guarantee the privacy of vehicle trajectory data, but still have some limitations.

In this paper, to the best of our knowledge, we take the first step to propose a differentially private vehicle trajectory data sanitization framework that can guarantee both strong privacy protection and high output utility. Differential privacy ensures the protection against inferences (whether any vehicle is involved in the input data) by the adversaries with arbitrary background knowledge. Our VTDP framework follows the sequential composition of multiple phases (parallel composition also exists in sampling phase II and phase III) but with limited overall bounds $(\epsilon + \epsilon', \delta)$. Our VTDP also greatly improves the output utility with the proposed vehicle trajectory interpolation based on the attributes of vehicle trajectory data. As validated in our experimental results, our VTDP framework generates fine-grained vehicle trajectory data with high utility, compared to the existing techniques (i.e., the VTL based techniques).

Acknowledgments

This work is partially supported by the National Science Foundation (NSF) under Grant No. CNS-1745894. The authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.
[2] B. Anandan and C. Clifton. Laplace noise generation for two-party computational differential privacy. In *PST* , pages 54–61, 2015.
[3] G. Acs, C. Claude and R. Chen. Differentially private histogram publishing through lossy compression. In *ICDM*, pages 1–10, 2012.
[4] X. Ban, R. Herring, P. Hao, and A. M. Bayen. Delay pattern estimation for signalized intersections using sampled travel times. *Transportation Research Record*, 2130(1):109–119, 2009.
[5] X. Ban, P. Hao, and Z. Sun. Real time queue length estimation for signalized intersections using travel times from mobile sensors. *Transportation Research Part C: Emerging Technologies*, 19(6):1133–1156, 2011.
[6] R. Bild, K. Kuhn, and F. Prasser. Safepub: A truthful data anonymization algorithm with strong privacy guarantees. In *PETS*, 2018(1):67–87, 2018.
[7] R. Chen, B. Fung, B. C. Desai, and N. M. Sossou. Differentially private transit data publication: a case study on the montreal transportation system. In *KDD*, pages 213–221, 2012.
[8] R. Chen, H. Li, A. K. Qin, S. P. Kasiviswanathan, and H. Jin. Private spatial data aggregation in the local setting. In *ICDE*, pages 289–300, 2016.
[9] Y. Cheng, X. Qin, J. Jin, and B. Ran. An exploratory shockwave approach to estimating queue length using probe trajectories. *J. Intellig. Transport. Systems*, 16(1):12–23, 2012.
[10] C. Cottrill and P. Thakuriah. Protecting location privacy: Policy evaluation. *Transportation Research Record: Journal of the Transportation Research Board*, (2215):67–74, 2011.
[11] F. Dotzer. Privacy issues in vehicular ad hoc networks. In *PETS*, pages 197–209, 2006.
[12] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Eurocrypt*, pages 486–503. Springer, 2006.
[13] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.
[14] H. Deng, I. King, and M. Lyu. Entropy-biased models for query representation on the click graph. In *SIGIRl*, pp. 339–346, 2009.
[15] M. Gotz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke. Publishing search logs – a comparative study of privacy guarantees. *TKDE*, 24(3):520–532, 2012.
[16] P. Hao, X. Ban, K. P. Bennett, Q. Ji, and Z. Sun. Signal timing estimation using sample intersection travel times. *IEEE ITSM*, 13(2):792–804, 2012.
[17] B. Hoh, T. Iwuchukwu, Q. Jacobson, D. B. Work, A. M. Bayen, R. Herring, J. C. Herrera, M. Gruteser, M. Annavaram, and J. Ban. Enhancing privacy and accuracy in probe vehicle-based traffic monitoring via virtual trip lines. *TMC*, 11(5):849–864, 2012.

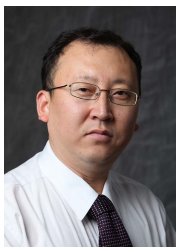
[18] Y. Hong, J. Vaidya, H. Lu, P. Karras, and S. Goel. Collaborative search log sanitization: Toward differential privacy and boosted utility. *TDSC*, 12(5):504–518, 2015.
[19] Y. Hong, J. Vaidya, H. Lu, and M. Wu. Differentially private search log sanitization with optimal output utility. In *EDBT*, pages 50–61. ACM, 2012.
[20] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. Herrera, A. Bayen, M. Annavaramb and Q. Jacobsonc Virtual trip lines for distributed privacy-preserving traffic monitoring. In *MobiSys*, pages 15–28, 2008.
[21] K. Jiang, D. Shao, S. Bressan, T. Kister, and K. Tan. Publishing trajectories with differential privacy guarantees. In *SSDBM*, pages 12:1–12:12, 2013.
[22] Y. Kanzawa, Y. Endo, and S. Miyamoto. Kl-divergence-based and manhattan distance-based semisupervised entropy-regularized fuzzy c-means. *JACIII*, 15(8):1057–1064, 2011.
[23] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. Random-data perturbation techniques and privacy-preserving data mining. *Knowl. Inf. Syst.*, 7(4):387–414, 2005.
[24] J. Le Ny, A. Touati, and G. J. Pappas. Real-time privacy-preserving model-based estimation of traffic flows. In *IEEE ICCPS*, pages 92–102, 2014.
[25] D. Leoni. Non-interactive differential privacy: a survey. In *ACM WOD*, pages 40–52, 2012.
[26] H. Li, H. Zhu, S. Du, X. Liang and X. Shen. Privacy leakage of location sharing in mobile social networks: Attacks and defense. *TDSC*, 15(4):646–660, 2018.
[27] M. Li, L. Zhu, Z. Zhang, and R. Xu. Achieving differential privacy of trajectory data publishing in participatory sensing. *Inf. Sci.*, 400(C):1–13, Aug. 2017.
[28] N. Li, W. Qardaji, and D. Su. On sampling, anonymization, and differential privacy: or, k-anonymization meets differential privacy In *ASIACCS*, pages 32–33, 2012.
[29] F. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD*, pages 19–30, 2009.
[30] F. McSherry and I. Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *KDD*, pages 627–636, 2009.
[31] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.
[32] T. P. Minka. Estimating a dirichlet distribution. Technical report, 2000.
[33] M. Mohammady. Differentially Private Event Stream Filtering with an Application to Traffic Estimation. PhD thesis, École Polytechnique de Montréal, 2015.
[34] M. Mohammady, L. Wang, Y. Hong, H. Louafi, M. Pourzandi, and M. Debbabi. Preserving Both Privacy and Utility in Network Trace Anonymization. In *CCS*, pages 459–474, 2018.
[35] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84, 2007.
[36] J. L. Ny, A. Touati, and G. J. Pappas. Real-time privacy-preserving model-based estimation of traffic flows. In *ACM/IEEE ICCPS*, pages 92–102, 2014.
[37] L. Ou, Z. Qin, S. Liao, Y. Hong and X. Jia. Releasing correlated trajectories: towards high utility and optimal differential privacy. *TDSC*, to Appear.
[38] N. Pham, R. K. Ganti, Y. S. Uddin, S. Nath, and T. Abdelzaher. Privacy-preserving reconstruction of multidimensional data maps in vehicular participatory sensing. In *European Conference on Wireless Sensor Networks*, pages 114–130, Springer, 2010.
[39] G. Poulis, S. Skiadopoulos, G. Loukides, and A. Gkoulalas-Divanis. Distance-based k^m-anonymization of trajectory data. In *MDM*, pages 57–62, 2013.
[40] N. Rizzo, E. Sprissler, Y. Hong, and S. Goel. Privacy preserving driving style recognition. In *ICCVE*, pages 232–237, 2015.
[41] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran. AMOEBA: robust location privacy scheme for VANET. *IEEE Journal on Selected Areas in Communications*, 25(8):1569–1589, 2007.
[42] R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *WPES*, pages 115–118, 2010.
[43] H. Wang, Y. Hong, Y. Kong and J. Vaidya. Publishing video data with indistinguishable objects. In *EDBT*, 2020.
[44] H. Wang, S. Xie, and Y. Hong. VideoDP: a universal platform for video analytics with differential privacy. *CoRR*, abs/1909.08729, 2019.
[45] B. Xu, X. J. Ban, Y. Bian, J. Wang, and K. Li. V2I based cooperation between traffic signal and approaching automated vehicles. In *IEEE IV*, pages 1658–1664, 2017.
[46] X. Yi and Y. Zhang. Equally contributory privacy-preserving k-means clustering over vertically partitioned data. In *Inf. Syst.*, 38(1):97–107, 2013.
[47] N. Zamzami and N. Bouguila. Text modeling using multinomial scaled dirichlet distributions. In *IEA/AIE*, pages 69–80, 2018.

Bingyu Liu received the B.Sc. degree from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013, and the M.Sc. degree from Illinois Institute of Technology, Chicago, IL, USA, in 2015. She is currently a Ph.D student in the Department of Computer Science at IIT. Her research interests include data privacy and security.

Shangyu Xie received the B.Sc degree from Shanghai Jiao Tong University with dual major in Electrical Engineering and Information Engineering, affiliated with the IEEE Honor Class in 2016. He is currently a Ph.D student in the Department of Computer Science at Illinois Institute of Technology, Chicago, IL, USA. His research focuses on data privacy and security.

Han Wang is currently a Ph.D student in the Department of Computer Science at Illinois Institute of Technology. She got her M.Sc degree from Huazhong University of Science and Technology. Her research interests include data privacy and security.

Yuan Hong (SM'18) received his Ph.D. degree in Information Technology from Rutgers, the State University of New Jersey. He is currently an Assistant Professor in the Department of Computer Science at Illinois Institute of Technology, Chicago, IL, USA. His research interests primarily lie at the intersection of privacy, security, optimization, and data mining. His research is supported by the National Science Foundation. He is a Senior Member of the IEEE.



Xuegang Ban received the B.Sc and M.Sc degrees in automotive engineering from Tsinghua University and Ph.D. degree in transportation engineering from the University of Wisconsin-Madison. He is currently an Associate Professor in the Department of Civil and Environmental Engineering, University of Washington. His research interests include transportation network system modeling and simulation, urban traffic modeling and control, and intelligent transportation systems.

Meisam Mohammady is a Ph.D candidate at the Concordia Institute for Information Systems Engineering, Concordia University. He got his M.Sc in Electrical Engineering at Ecole Polytechnique Montreal and his B.Sc in Electrical and Computer Engineering from Sharif University Of Technology. His research interests include privacy, computational learning theory, and applied mathematics.