Synthesizing Plausible Privacy-Preserving Location Traces

Vincent Bindschaedler
UIUC
bindsch2@illinois.edu

Reza Shokri
Cornell Tech
shokri@cornell.edu

***Abstract*— Camouflaging user's actual location with fakes is a prevalent obfuscation technique for protecting location privacy. We show that the protection mechanisms based on the existing (ad hoc) techniques for generating fake locations are easily broken by inference attacks. They are also detrimental to many utility functions, as they fail to credibly imitate the mobility of living people. This paper introduces a systematic approach to synthesizing plausible location traces. We propose metrics that capture both geographic and semantic features of real location traces. Based on these statistical metrics, we design a privacy-preserving *generative* model to synthesize location traces which are plausible to be trajectories of some individuals with consistent lifestyles and meaningful mobilities. Using a state-of-the-art quantitative framework, we show that our synthetic traces can significantly paralyze location inference attacks. We also show that these fake traces have many useful statistical features in common with real traces, thus can be used in many geo-data analysis tasks. We guarantee that the process of generating synthetic traces itself is privacy preserving and ensures plausible deniability. Thus, although the crafted traces statistically resemble human mobility, they do not leak significant information about any particular individual whose data is used in the synthesis process.**

I. Introduction

> It is preferable not to travel with a dead man.
>
> Henri Michaux

A popular method to protect the privacy of a mobile user, who queries a location-based service (LBS), is to hide her true query among fake queries. Users keep the obtained information to their real queries and discard the responses to all their fake queries. The existing techniques to generate fake locations [10], [23], [24], [26], [45], [49], [56], are based on very simple heuristics such as i.i.d. location sampling and sampling locations from a random walk on a grid or on the road network or between points of interest. The generated location traces using these types of techniques fail to capture the essential semantic and even some basic geographic features of the mobility of a *living* human who has a consistent lifestyle and meaningful mobility. Thus, as we quantitatively show, they poorly protect users' privacy against location inference attacks that can easily filter out the trajectories of the jumping *dead*.

In order to be plausible, synthetic traces need to statistically resemble real traces, thus themselves should be generated in a privacy-preserving manner. Consider the most naive protection scheme where location samples from the trajectory of Alice (a real person who perhaps is not even using the LBS) are used to mask locations of Bob (a LBS user). Clearly this is too intrusive with respect to the privacy of Alice, although it confuses the attacker about Bob's truly visited locations. The obfuscation techniques that compose an area of fake locations around the user's true location, e.g., [30], [39], [52], are inappropriate for similar reasons: they have a strong statistical correlation with the user's true trace and do not introduce much error to location inference attacks. Thus, they are less privacy preserving than even, for example, i.i.d generated fake locations [45].

In this paper, we present and evaluate the first formal and systematic methodology to generate fake yet semantically real privacy-preserving location traces. In this approach, we propose two mobility *metrics* that capture how realistic a synthetic location trace is with respect to the geographic and semantic dimensions of human mobility. We then construct a probabilistic *generative model* that produces synthetic yet plausible traces according to these metrics. We build our generative model upon a dataset of real location traces as *seeds*. Thus, the model itself needs to be privacy-preserving. To this end, we design *privacy* tests to control and limit information leakage about the seed dataset. We then use state-of-the-art location inference *attacks* to evaluate the effectiveness of our fake traces in preserving the privacy of LBS users. On a set of real location traces, we show that the attacker's probability of error [46] in estimating the true location of users over time is 0.9972 when using our method, i.e., we achieve close to maximum privacy. By comparison, the attacker's error is 0.2958, 0.3066, 0.3802, and 0.7486 when using existing i.i.d. sampling and random walk methods.

Our scheme is based on the fact that mobility patterns of different individuals are semantically similar, regardless of which geographic locations they visit. These common features of human mobility stem from their similar lifestyles, e.g., traversing between home, workplace, friends' place, favorite shops and recreational places, and occasional new locations. These mobility patterns share a similar structure that reflect the general behavior of a population (at a high level [48]). We model the mobility of each individual in two dimensions: *geographic* and *semantic*. In addition to the common mobility patterns (i.e., how people move in a city), the geographic features are mostly specific to each individual (e.g., what everyone refers to as her "home" is located in a geographically distinct place), whereas the semantic features are mostly generic and representative of overall human mobility behavior (e.g., most people have a "home" where they stay overnight). Thus, the semantic representations of human mobilities are very similar, especially within a culture group with similar life

styles. We extract these common semantic features as well as the aggregate geographic features from real mobility datasets, as seed. Using this, we probabilistically generate synthetic traces which are geographically probable and semantically similar to real traces. This results in a set of traces for nonexistent individuals with meaningful lives and consistent mobility patterns as any real individual in the seed dataset.

We preserve the privacy of seed traces. Our first step is a random and independent sampling of the seeds, which is shown to be very effective in boosting the privacy of individuals in a database [16]. We generate synthetic traces from sampled seed traces. We then accept a synthetic trace only if (1) it is geographically dissimilar to seeds and (2) the same synthetic trace could have been generated from $k \geq 1$ non-sampled alternatives. This ensures *plausible deniability*. This is intrinsically similar to the notions of crowd blending privacy [16], zero knowledge privacy [17], and outlier privacy [31]. Our privacy guarantees protect the privacy of seed traces against the following threats: *inference* attacks (to learn which locations the seed contributors have visited), and *membership inclusion* attack (to learn if a particular individual with certain semantic habits has been in the seed dataset).

The application of our generated synthetic traces goes beyond protecting the privacy of mobile users in location-based services. Our generated traces can also be used for a variety of geo-data analysis tasks such as modeling human mobility [28], map inference [29], points of interest extraction [59], semantic annotation of locations [55], and location optimization for opening new shops [22]. We list six features of traces that contribute to these applications and show that our synthetic traces exhibit a similar performance to what is achieved from real traces on these tasks. For example, out of 400 locations, the accuracy of synthetic traces in extracting the top-35 points of interest is 96.7% compared to real traces. This is 88.5% and 100% respectively for the top-30 and top-40 points of interest.

Novelty. We design, implement as a tool, and evaluate with real data the first formal privacy-preserving generative model to synthesize plausible location traces. Our privacy guarantees ensure plausible deniability for individuals whose trace is used by our algorithm. In a *LBS scenario*, we show that our fake traces can bring near maximum location privacy (against state-of-the-art inference attacks) for the users with minimum overhead (i.e., the number of fake locations needed to be sent to the LBS server along with the true location). We also show that these traces do not perturb the semantic profile of the user in location-based recommender systems. In the *dataset release scenario*, where only synthetic traces are released for analysis, we show that useful features are preserved for multiple geo-data analysis tasks. We design privacy tests to ensure that the synthetic traces do not leak more information about the real traces from which they are generated than alternative traces.

II. Related Work

Synthetic (also called fake or dummy) information can protect privacy and security in many different systems such as web search [18], [20], anonymous communications [5], [12], authentication systems [21], and statistical analysis [33], [42]. In all these scenarios, the main challenge and the still open problem is to generate context-dependent fake information that resembles genuine user-produced data and also provides an acceptable level of utility while enhancing privacy of users.

In location-based services, location obfuscation is a prevalent non-cryptographic technique to protect location privacy. It does not require changing the infrastructure, as it can also be done entirely on the user's side either by altering (perturbing) the location coordinates to be reported or by sending fake location reports interleaved or along with the true locations.

Many location perturbation techniques are based on adding some noise to the user's location coordinates or reducing its granularity, e.g., [3], [4], [19], [47]. The downside of these techniques is that they reduce the users' experienced LBS service quality. This is because the server provides contextual information related to the queried location and not the true location of the user. So, users have to trade service quality for privacy. Optimal solutions for location perturbation are proposed [7], [47] which show the high cost of this technique.

Hiding the user's true location among fake locations is a promising yet not systematically-approached method to protecting location privacy. There are few simple techniques proposed so far: adding independently selected locations drawn from the population's location distribution [45], generating dummy locations from a random walk on a grid [24], [56], constructing fake driving trips by building the path between two random locations on the map given the more probable paths traveled by drivers [26], adding noise to the paths generated by road trip planner algorithms [10], or generating the path between points of interests [49] and pausing at those points [23]. All these solutions lack a formal model for human mobility and do not consider the semantics associated with location traces. Thus, the generated traces can be easily distinguished from real trajectories, as we show in this paper.

To address potential misunderstandings, we contrast our work with *anonymization*, and releasing *aggregate statistics*. (1) Anonymization consists in removing identifiers of individuals in the data, and publishing only the resulting (sanitized) dataset. While this preserves utility, it does not provide much privacy protection. Indeed, many researchers have shown that anonymous traces are easily re-identifiable [32], [50], [58]. (2) Releasing privacy-preserving aggregate statistics has been proposed in many contexts. In particular, there has been a lot of recent work on releasing differentially-private histograms for various types of data [1], [2], [54]. These are totally unsuitable to be used in the LBS scenario and, in general, in applications which require *full* location traces. For example, to obtain a full location trace from a private histogram, one way is to repeatedly and independently sample locations from it. This results in an unlikely trace which include "jumps" between locations regardless of their distance and mobility constraints. In particular, [1] considers the problem of releasing differentially-private location histograms at various time intervals. Also, [9] releases variable length n-grams with differential privacy guarantee, which cannot produce full location traces.

In summary, the existing approaches do not evaluate how plausible and privacy-preserving their synthesized traces are. They are only based on simple heuristics about human mobility. Hence, they do not properly preserve geographic features of it, and completely ignore its semantic features. As a result, their produced traces are not suitable in many scenarios. They also lack privacy guarantees (and fail) against inference attacks. This paper fixes these shortcomings and enables us to reason about and generate plausible synthetic location traces.

There are also several notable related works which appear similar to this work but have subtle and important differences. An example is DP-WHERE [36] which uses Call Detail Records (CDRs) databases to produce differentially-private synthetic databases with a distribution close to real CDRs. However, CDRs are not equivalent to *full* location trajectories because the location is only known at the time when a call is made. Another example, is wPINQ [40] which achieves differential privacy by calibrating down the weight of some data records. wPINQ further proposes a way of generating synthetic datasets using Markov chain Monte Carlo methods. The techniques used, scenarios, and utility evaluation prohibit a direct comparison with our work: wPINQ focuses on graphs given noisy measurements about the number of triangles, whereas we consider the problem of generating plausible full location trajectories.

Finally, Dwork et al. [15] introduce a class of mechanisms called Propose-Test-Release (PTR) which first picks a bound on the sensitivity (of a statistic of interest) and then (privately) tests whether noise calibrated to this candidate bound is sufficient to ensure differential privacy. If so, then a noised output is released, otherwise no output is produced. There are two major differences between PTR and our work. First, we aim to generate synthetic location traces, whereas [15] seeks privacy-preserving ways to estimate robust statistics such as discovering the median of a dataset without prior knowledge of the scale of the data. Second, the PTR framework performs a test of the *sensitivity* of a statistic before deciding to release a noised output, whereas our privacy tests are there to test the synthetic traces generated themselves before deciding whether to release them.

III. Our Scheme

In this section, we present a sketch of, and describe the main intuition behind our scheme for generating fake traces. We assume that time and space are discrete, so a location trace is represented as a sequence of visited locations over time. In our scheme, we synthesize a trace through a multiple step process. We transform a (geographic) seed trace into the semantic space and probabilistically transform it back to the geographic space. Thus, the sampled trace is geographically and semantically plausible. Figure 1 illustrates our scheme.

A. Subsampling the Seeds

We generate synthetic location traces by using a set of real traces, from which we randomly subsample a set which we refer to as the *seed dataset*. We refer to the set of traces that are not sampled as the *alternative dataset*. The reason for subsampling becomes more clear when we explain our privacy guarantees. Put simply, to guarantee plausible deniability, we ensure that there are k alternative traces that could have produced a similar synthetic trace generated from a seed.

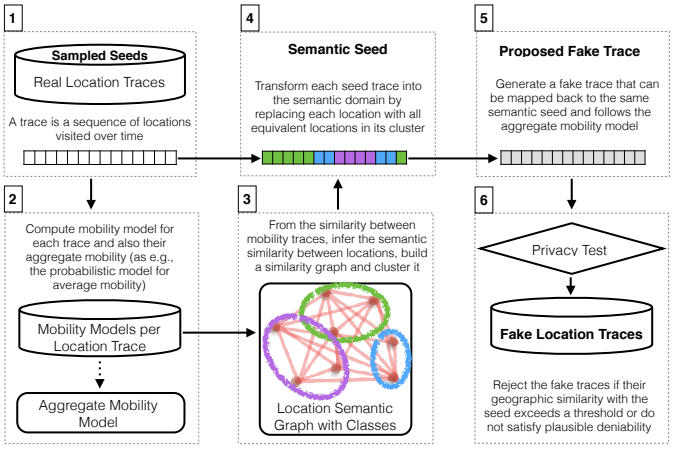


Fig. 1: Sketch of the proposed scheme.

B. Computing the Semantic Similarity

Our goal here is to compute the semantic similarity between locations. To this end, we start with modeling mobility of seed locations. For each trace (i.e., sequence of locations visited in the trace) in the seed dataset, we compute a probabilistic *mobility model* that represents the visiting probability to each location and transition probability among the locations (see Section IV-A). The mobility model encompasses the spatiotemporal behavior of each individual across different locations. Time, duration, and probability of visiting a location, as well as the probable previous and subsequent locations are computable from the mobility model.

We analyze and discover the semantic relation between different locations in a consistent manner by considering all locations together. To this end, we propose a *semantic similarity metric* (see Section IV-D). Intuitively, we assign a higher similarity value to a pair of locations if multiple individuals have similar spatiotemporal activities in them. We find the optimal way to map the visited locations in a pair of traces such that the mapping maximizes the statistical similarity between their mobility models. The semantic similarity metric is therefore the statistical similarity between mobility models under the optimal semantic mapping between locations. This means that if we were to translate the locations visited by two individuals according to the discovered best mapping, they would follow the same mobility model when their semantic similarity is high (i.e., have similar life styles). For example, consider Alice and Bob spending all day at their respective work locations w_A and w_B, and all night at their respective home locations h_A and h_B. Obviously, their mobility models are semantically very similar, although it might be the case that $h_A \neq h_B$ and $w_A \neq w_B$. In this example, the best semantic mapping between locations will be $w_A \leftrightarrow w_B$ and $h_A \leftrightarrow h_B$.

That said, the semantic similarity metric we propose goes beyond simply finding the best mapping for home and work. Indeed, the best mapping is over all locations, so it may be that Alice's favorite bar is mapped to Bob's favorite nightclub, if Alice and Bob visit those places in a similar way.

For each pair of mobility models of traces in the seed dataset, we compute their semantic similarity as well as the best semantic mapping between their locations. Note that the semantic similarity is quantifying the similarity of two mobility models, not that of two location traces. This incorporates the similarity between statistical information in the traces rather than their exact sequence of locations. We then aggregate all the location matchings across all seed trace pairs, with weights based on the semantic similarity between mobility models, and construct a *location semantic graph*, where the nodes are locations and the weight of the edges is the average semantic similarity between the locations over the dataset.

C. Forming Location Semantic Classes

The location semantic graph enables us to infer which locations have similar meanings (or purpose) for different people. The locations that have higher semantic similarity can be grouped together to represent one *location semantic class*. We run a clustering algorithm on the location semantic graph to partition locations into distinct classes. Regardless of their geographic positions, the locations that fall into the same class are visited in the same way by different people. In other words, their visit probability, time of visit, and the probabilities of transition from/to them to/from other locations with the same type is similar. Thus, we can consider them as being semantically equivalent. So, using the notations of our previous example, w_A and w_B should belong to the same cluster that can represent "workplace" locations, and h_A and h_B should be grouped into another cluster representing residential or "home" locations.

D. Synthesizing a Trace

We use the location semantic classes as the basis to generate synthetic traces. In addition to being semantically realistic, the fake traces must be geographically consistent with the general mobility of individuals in the considered area. For example, the speed of moving between locations and the duration time of staying in a location depend on the time of the day or the probabilities of different paths that cross those locations. To capture these patterns, we compute an *aggregate mobility* model from the traces in the seed dataset by averaging their corresponding mobility models.

The goal is to synthesize traces that are semantically similar to real traces. To this end, our algorithm starts with a *seed* trace and converts it to a probabilistically generated semantically similar synthetic trace which is consistent with the aggregate mobility model. We first *transform* the geographic seed trace into the semantic domain, then we use the transformed semantic trace to *sample* from the domain of all geographic traces that could have been transformed to the same semantic trace. The transformation and sampling procedures, which are at the heart of this step, are done as follows.

In the *transformation* process, we replace the geographic locations in the seed with the locations that are in the same semantic class. This *semantic trace* is a sequence of location sets. For *sampling* a synthetic trace, we address the following problem. We want to construct a trace that follows the aggregate mobility model under the constraint that its locations over time are a subset of locations of the seed semantic trace. Hence, both the synthetic trace and the seed trace can be transformed to the same semantic trace. This makes the synthetic trace semantically plausible. We add some randomness to the locations in the semantic trace to increase the flexibility of our algorithm. Many methods can be used to sample the fake trace that satisfies our constraints. We make use of dynamic programming algorithms that construct the traces efficiently (see Section V).

We can repeatedly generate synthetic traces from each seed trace in the dataset, each of which having a probability according to the aggregate mobility model. After generating each trace, however, we need to make sure that it is not geographically similar to the seed trace. This is because we do not want to leak information about the real seed trace. To this end, we add a test to compute the *geographic similarity* between the seed trace and the fake trace to *reject* the sample traces that are more similar than a threshold to the seed trace. Thus, we make sure that the semantically similar synthetic traces are indeed geographically dissimilar to the traces in our dataset, hence do not leak information about visited locations in the real traces. We also ensure that the semantics of a synthetic trace do not leak about a seed trace more than what they leak about alternative traces (which are not among the seeds and our algorithm is independent of). To this end, we run the *plausible deniability* privacy test (see Section V).

IV. Mobility Similarity Metrics

In this section, we present a probabilistic model for mobility, and propose two metrics to analyze the geographic and semantic similarity between two mobility models. Table I presents the list of notations that we use in this paper.

A. Mobility Model

We model the user mobility as a time-dependent first-order Markov chain on the set of regions (locations). As users have different activities and mobility patterns during different periods of time, we assume that time is partitioned into time periods, e.g., morning - afternoon - evening - night. So, the mobility profile $\langle p(u), \pi(u) \rangle$ of a given user u is a *transition* probability matrix of the Markov chain associated with the user's mobility (from a region to another), and the user's *visiting* probability distribution over the regions, respectively. Note that these probabilities are dependent on each other, and together they constitute the joint probability of two regions that are subsequently visited by the user. The entry $p^{r'}_{r,\tau,\tau'}(u)$ of $p(u)$ is the probability that user u will move to region r' in the next time instant (which will be in time period τ'), given

$\mathcal{R}$	Set of locations
R	Number of locations
r	A location
$\mathbf{r}$	Random variable associated with a location
T	Number of time periods
τ	A time period
$p(u)$	Transition probability matrix of user u
$\pi(u)$	Visiting probability vector of user u
$\langle p(u), \pi(u) \rangle$	Mobility profile of user u
$p^{\mathbf{x}}_{\mathbf{y}}(u)$	Probability of $\mathbf{x}$ given $\mathbf{y}$ according to u's mobility model
$d(\cdot)$	A distance function (between locations)
$M_d(p, q)$	Mallows distance between probability distributions p and q based on a distance function $d(\cdot)$
σ	A permutation function
$\mathsf{sim_G}(u, v)$	Geographic similarity between mobility of u and v
$\mathsf{sim_S}(u, v)$	Semantic similarity between mobility of u and v
σ^v_u	Optimal semantic mapping between locations of u and v
$\mathcal{S}$	Set of real traces used as seeds to generate synthetic traces
$\mathcal{A}$	Set of alternative real traces used in plausible deniability test
$\langle \bar{p}, \bar{\pi} \rangle$	Aggregate mobility model
$\mathcal{C}$	A partition on $\mathcal{R}$, representing location semantic classes. $\mathcal{C}_i$ is the set of locations in class (partition) i
$\mathcal{F}$	A set of fake locations generated from $\mathcal{S}$

TABLE I: Table of notations

that she is now (in time period τ) in region r. The entry $\pi^r_\tau(u)$ is the probability that user u is in region r in time period τ. We can compute $\pi(u)$ from traces or directly from $p(u)$ (in some circumstances). Let the random variable $\mathbf{A}^t_u$ represent the actual location of user u at time t, and τ^t be the time period associated with $\mathbf{A}^t_u$. So, the mobility profile of a given user u consists of the following probabilities:

$$p^{r'}_{r,\tau,\tau'}(u) = \Pr\{\mathbf{A}^{t+1}_u = r' \mid \mathbf{A}^t_u = r; \tau^{t+1} = \tau', \tau^t = \tau\},$$
$$\pi^r_\tau(u) = \Pr\{\mathbf{A}^t_u = r; \tau^t = \tau\} \quad (1)$$

This Markovian model can predict the location of an individual to a great extent, as it takes both location and time aspects into account. It can become even more precise, by increasing its order, or by enriching its state. Our framework can incorporate new dimensions similar to the way we model the time periods. To learn the probabilities of the mobility profile (1), from location traces, we can use maximum likelihood estimation (if the traces are complete) or make use of algorithms such as Gibbs sampling (if the traces have missing locations or are noisy) [46].

B. Mobility Similarity Metrics

We propose two metrics to compare the mobility of two users and compute their similarities: *geographic* and *semantic* similarity. In this subsection, we describe the intuition behind these metrics, and in the following subsections, we formally define and provide the algorithms to compute them.

The *geographic similarity* metric captures the correlation between location traces that are generated by two mobility profiles. It reflects if two users visit similar locations over time with similar probabilities and if they move between those locations also with similar probabilities. Using this metric, for example, two individuals who live in the same region A and their workplace is in the same region B potentially have very similar mobilities, as they spend their work hours in B and most of their evenings in A.

The geographic similarity between the mobility models of two random individuals is usually low. However, if we ignore their exact visited locations, they tend to share similar patterns for visiting locations with similar semantics (locations therein they have similar activities). Consider the semantic dimension of locations as a coloring of them on the map. Besides the geographic correlation between location traces, we can compute their correlation at the semantic level too (by reducing the set of locations to colors and computing the similarity of colored traces). This is the intuition behind our *semantic similarity* metric. In this case, if the pair of locations that two individuals visit over time have the same semantic, their mobility models are also semantically similar (even if they do not intersect geographically). For example, if we transform trace X by replacing its locations with their corresponding semantically similar locations in trace Y, the transformed trace becomes statistically similar to Y. So, two traces are semantically similar if their locations can be mapped (translated) to each other in this way.

C. Geographic Similarity Metric

We define this similarity metric based on the Earth Mover's Distance (EMD) for probability distributions. The EMD is widely used in a range of applications [43], [44], and can be understood by thinking of the two distributions as piles of dirt where it represents the minimum amount of work needed to turn one pile of dirt (i.e., one distribution) into the other; the cost of moving dirt being proportional to both the amount of dirt and the distance to the destination. The special case of EMD for probability distributions has been shown to be equivalent to the Mallows distance [27].

Let $\mathbf{X}$ and $\mathbf{Y}$ be discrete random variables with probability distributions p and q, such that $\Pr\{\mathbf{X} = x_i\} = p_i$ and $\Pr\{\mathbf{Y} = y_i\} = q_i$, respectively, for $i = 1, 2, \ldots, n$. We also have $\sum_i p_i = 1$ and $\sum_i q_i = 1$.

Definition 1. *(From [27]) Let $d(\cdot)$ be an arbitrary distance function between $\mathbf{X}$ and $\mathbf{Y}$. The* Mallows distance $M_d(p, q)$ *is defined as the minimum expected distance between $\mathbf{X}$ and $\mathbf{Y}$ with respect to $d(\cdot)$ and to any joint distribution function f for $(\mathbf{X}, \mathbf{Y})$ such that p and q are the marginal distributions of $\mathbf{X}$ and $\mathbf{Y}$, respectively.*

$$M_d(p, q) = \min_f \{\mathbb{E}_f[d(\mathbf{X},\mathbf{Y})] : (\mathbf{X},\mathbf{Y}) \sim f, \mathbf{X} \sim p, \mathbf{Y} \sim q\}, \quad (2)$$

where the expectation, minimized under f, is

$$\mathbb{E}_f[d(X, Y)] = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij}\ d(x_i, y_j). \quad (3)$$

In addition to the two constraints $\sum_{i=1}^n \sum_{j=1}^n f_{ij} = 1$ and $f_{ij} \geq 0$, for all i, j, the joint probability distribution function f must also satisfy $\sum_{i=1}^n f_{ij} = q_j$ and $\sum_{j=1}^n f_{ij} = p_i$.

Note that, for given p and q, the minimum f is easily computed by expressing the optimization problem as a linear program.

Using the previous definition, we define the geographic similarity metric based on the Mallows distance.

Definition 2. *Let $d(\cdot)$ be an arbitrary distance function. The* dissimilarity *between two mobility profiles $\langle p(u), \pi(u)\rangle$ and $\langle p(v), \pi(v)\rangle$ (belonging to individuals u and v), is defined as the expected Mallows distance of the next random locations $\mathbf{r}'$ and $\mathbf{r}''$ according to the mobility profiles of u and v, respectively. More formally, it is*

$$\mathbb{E}_{(u)}[M_d(p^{\mathbf{r}'}_{\mathbf{r},\tau,\tau'}(u), p^{\mathbf{r}''}_{\mathbf{r},\tau,\tau'}(v))], \tag{4}$$

where $p^{\mathbf{r}'}_{r,\tau,\tau'}(u)$ and $p^{\mathbf{r}''}_{r,\tau,\tau'}(v)$ denote the conditional probability distributions of the next location, given the current location and the current and next time periods. The Mallows function is computed over random variables $\mathbf{r}'$ and $\mathbf{r}''$, and the expectation is computed over random variable $\mathbf{r}$ and time periods τ and τ'.

We define the geographic similarity *between mobility patterns of u and v as*

$$\mathsf{sim}_{\mathsf{G}}(u, v) = 1 - \frac{\mathbb{E}[M_d(p^{\mathbf{r}'}_{\mathbf{r},\tau,\tau'}(u), p^{\mathbf{r}''}_{\mathbf{r},\tau,\tau'}(v))]}{z_g}, \tag{5}$$

where z_g is a normalization constant equal to the maximum value of (the expectation of) the Mallows distance given $d(\cdot)$, ensuring that the geographic similarity always lie in $[0, 1]$.

We compute the geographic *dissimilarity* using the law of total expectation. This also clarifies its meaning by showing more directly the role of the random variables.

$$\begin{aligned}&\mathbb{E}[M_d(p^{\mathbf{r}'}_{\mathbf{r},\tau,\tau'}(u), p^{\mathbf{r}''}_{\mathbf{r},\tau,\tau'}(v))]\\ &= \sum_{r,\tau,\tau'} M_d(p^{\mathbf{r}'}_{r,\tau,\tau'}(u), p^{\mathbf{r}''}_{r,\tau,\tau'}(v)) \cdot p^{r,\tau,\tau'}(u).\end{aligned} \tag{6}$$

This is simply the average, for each time and location, of the EMD between the distributions of the next location of u and v. So, for each current location (and time), we use the EMD to compute the dissimilarity between the distributions representing the next locations of users u and v, respectively. The current location is taken according to user u's mobility profile, making this definition asymmetric.

For a particular distance function $d(\cdot)$, the Mallows distance definition can be expanded and previous expressions can be further simplified. This is the case for $d(i, j) = \mathbb{1}_{i\neq j}$, for which $M_d(p, q)$, for arbitrary probability distributions p and q, has closed form $1 - \sum_i \min\{p_i, q_i\}$.

Using the dissimilarity metric, we can compute the *geographic similarity* between the mobility profiles $\langle p(u), \pi(u)\rangle$ and $\langle p(v), \pi(v)\rangle$, for any distance function (e.g., hamming distance, Euclidean distance). For example, considering hamming distance $d(r, r') = \mathbb{1}_{r\neq r'}$, the geographic similarity is:

$$\sum_{r,r',\tau,\tau'} p^{\tau'}_{r,\tau}(u)\pi^{r,\tau}(u) \min\{p^{r'}_{r,\tau,\tau'}(u), p^{r'}_{r,\tau,\tau'}(v)\}. \tag{7}$$

We emphasize that this definition leads to an asymmetrical similarity measure, i.e., the similarity of u to v need not be the same as the similarity of v to u. In principle, this metric can also be computed using measures other than EMD. For example, one can use Kullback-Leibler divergence measure [11] to compute the difference between two probability distributions, ignoring the distance between the locations. We emphasize that we use EMD, in our geographic similarity metric, as we also want to include the distance function $d(\cdot)$ between locations in computing the difference between two mobility models.

Consider now the computation of the geographic similarity. For the case, $d(r, r') = \mathbb{1}_{r\neq r'}$, the computation according to closed-form of (7) takes $O(T^2 \cdot R^2)$ operations, where T is the number of time periods and R is the number of locations (regions). For arbitrary $d(\cdot)$ with no closed-form expressions, the geographic similarity is obtained through $T^2 \cdot R$ EMD computations. Each of these EMD computations involves minimizing the Mallows distance, that is equivalent to solving the linear program given by (2).

D. Semantic Similarity Metric

The semantic similarity metric builds upon the basic assumption that for two individuals u and v there exists an (unknown) semantics mapping σ of locations $\mathcal{R}$ onto itself (i.e. a permutation) such that $\mathcal{R}$ for u, and $\sigma^{-1}(\mathcal{R})$ for v semantically match. It is important to note that assuming such a mapping does not commit us to trying to learn it based on modeling location semantics directly. Instead, we define the *hidden* semantic similarity between u and v as the maximum geographic similarity taken over all possible mappings σ. We define semantic similarity metric as follows.

Definition 3. *Let σ be the mapping of locations of u to locations of v. Let $\mathbf{r}$, $\mathbf{r}'$, and $\mathbf{r}''$ be random variables for locations, and τ and τ' be two time periods. We define the semantic* dissimilarity *between u and v for moving in the sequence of time periods $\{\tau, \tau'\}$ as*

$$\mathcal{D}^v_u(\{\tau, \tau'\}) = \min_{\sigma} \mathbb{E}\left[M_d(p^{\mathbf{r}'}_{\mathbf{r},\tau,\tau'}(u), p^{\sigma(\mathbf{r}'')}_{\sigma(\mathbf{r}),\tau,\tau'}(v))\right], \tag{8}$$

where the Mallows distance $M_d(\cdot)$ is computed over the random variable $\mathbf{r}'$ and the expectation is computed over the random variable $\mathbf{r}$ given time periods τ and τ'.

Now, we define the semantic similarity *between u and v over any sequences of time periods as*

$$\mathsf{sim}_{\mathsf{S}}(u, v) = 1 - \frac{\mathbb{E}\left[\mathcal{D}^v_u(\{\tau, \tau'\})\right]}{z_s}, \tag{9}$$

where z_s is a normalization constant equal to the maximum value of (the expectation of) the Mallows distance given $d(\cdot)$, ensuring that the semantic similarity always lie in $[0, 1]$.

What we compute in (8) is the minimum geographic mobility dissimilarity between u and v where the locations of v are relabeled and mapped to locations of u according to the permutation function σ^v_u (which is the σ that minimizes 8). The intuition is the following. Consider two individuals u and v are at $\mathbf{r}$ and $\sigma(\mathbf{r})$, respectively, at time period τ. The Mallows distance M_d computes how dissimilar their movement will be to the next location which are represented with random variables $\mathbf{r}'$ for u and $\sigma(\mathbf{r}'')$ for v. If, according to a mapping,

the way that they move between these locations is similar, they behave similarly with respect to those locations. If this is true across different time periods and location pairs, their mobilities are similar. So, the semantic similarity between two individuals is determined by σ_u^v.

We compute this metric at two different levels of accuracy of the mobility model. If we only consider the visiting probability π part of each individual's mobility profile, we compute $\mathsf{sim_S}$ as follows: Let us consider the hamming distance function $d(r, r') = 1_{r \neq \sigma^{-1}(r')}$. In this case, we can compute the semantic similarity metric as

$$1 - \sum_{\tau} \Pr\{\tau\} \max_{\sigma} \sum_{r} \min\{\pi_\tau^r(u), \pi_\tau^{\sigma(r)}(v)\}. \quad (10)$$

Note that the computation of (10) requires finding the mapping σ which maximizes the inner term for each time period τ. Since there are $R!$ possible candidates for the maximum mapping σ, a brute-force approach is inefficient. However, the problem's structure resembles that of a linear assignment. Focusing on the inner sum, we see that each term (each r) can be associated with R values of $\sigma(r)$ independently of the other components of σ. To recast the problem as a linear assignment, we construct a bipartite graph where the nodes represent $\mathcal{R}$ and $\sigma(\mathcal{R})$, and each edge represents the association (through σ) of r with $\sigma(r)$. The maximum weight assignment of the constructed bipartite graph gives the permutation σ. The running time of this procedure is $O(T \cdot R^3)$ using the Hungarian algorithm [37].

In the case where we consider the more accurate mobility profile $\langle p, \pi \rangle$, it can be computed as follows:

$$1 - \sum_{\tau,\tau'} \max_{\sigma} \sum_{r,r'} \pi^{r,\tau}(u) p_{r,\tau}^{\tau'}(u) \min\{p_{r,\tau,\tau'}^{r'}(u), p_{\sigma(r),\tau,\tau'}^{\sigma(r')}(u)\}. \quad (11)$$

It is not known whether there is an efficient algorithm to compute the semantic similarity according to (11). The difficulty comes from having to consider assignments of pairs: (r, r') to $(\sigma(r), \sigma(r'))$, which makes this computation look similar to the Quadratic Assignment Problem (QAP) [38], known to be NP-Hard and APX-Hard. But, (11) can be approximated using e.g., the Metropolis-Hastings algorithm [35] which we use in the case of considering both visiting and transition probabilities (see [34] for details), or Simulated Annealing [8].

V. Synthesizing Location Traces

In this section, we present the details of our algorithms for generating synthetic traces. See also Figure 2. Note that the processes of generating and using fake traces are completely separate. When a set of traces is synthesized, they can be used in different scenarios accordingly.

A. Transforming Traces into Semantic Domain

We sample a dataset $\mathcal{S}$ from some real traces. We use traces in $\mathcal{S}$ as seeds to generate synthetic traces. Each seed trace in the dataset comes from a different individual. Generating a fake trace starts by transforming a real trace (taken as seed) to a semantic trace. To this end, we require to know the semantic coordinates of the seed trace. We compute the semantic similarity between all locations in $\mathcal{R}$, and create a location semantic graph $G\langle \mathcal{R}, E, w \rangle$ such that the vertices are in $\mathcal{R}$ and the weight $w_G(r, r')$ on the edge between locations r and r' is the weighted sum of the number of pairs of users u and v for whom r and r' are semantically mapped (i.e., $r = \sigma_u^v(r')$), weighted according to their similarity. Then, we create the equivalent semantic classes $\mathcal{C}$ by running a clustering algorithm on this graph. We make use of the k-means clustering algorithm, and we choose the number of clusters such that it optimizes the clustering objective. We present the sketch of this algorithm in Figure 2 – SemanticClustering().

We then convert the seed location trace $seed$ into its corresponding semantic trace $semseed$ by simply replacing each location in the trace with all its semantically equivalent locations (according to the semantic classes $\mathcal{C}$). Figure 3 depicts an example of such a semantic seed. Intuitively, this composite trace encompasses all possible geographic traces that are semantically similar to the original seed trace. To be more flexible with respect to the traces that we can generate,

SemanticSimilarity(u, v)

Compute mobility models $\langle p(u), \pi(u) \rangle$ and $\langle p(v), \pi(v) \rangle$
Compute optimal mapping σ_u^v from (8)
Compute semantic similarity $\mathsf{sim_S}(u, v)$ from (9)
Return $\mathsf{sim_S}(u, v), \sigma_u^v$

SemanticClustering$(\mathcal{R}, \mathcal{S}, \kappa)$

Initiate weighted graph G with locations $\mathcal{R}$ as vertices
Forall $u, v \in \mathcal{S}, u \neq v$:
  Let $s_u^v, \sigma_u^v \leftarrow$ SemanticSimilarity(u, v)
  Forall locations $r, r' \in \mathcal{R}$ such that $r' = \sigma_u^v(r)$:
    Let edge weight $w_G(r, r') \leftarrow w_G(r, r') + s_u^v$
Let $\mathcal{C} \leftarrow$ K-Means(G, κ)
Return $\mathcal{C}$

PrivacyTest$(fake, seed, \mathcal{A}, \delta_s, \delta_i, \delta_d)$

Let geographic similarity $sim \leftarrow \mathsf{sim_G}(fake, seed)$
Let intersection between fake and seed $int \leftarrow |\text{intersection}(fake, seed)|$
Let $pl \leftarrow TRUE$ if
  $\exists a_1, \cdots, a_k \in \mathcal{A}$ s.t. $\forall i, |\mathsf{sim_S}(seed, fake) - \mathsf{sim_S}(a_i, fake)| \leq \delta_d$
Return TRUE if $int \leq \delta_i$ and $sim \leq \delta_s$ and pl

Synthesizer$(\mathcal{R}, \mathcal{S}, \mathcal{A}, all\ parameters)$

Let aggregate mobility model $\langle \bar{p}, \bar{\pi} \rangle$ be average of $\langle p(u), \pi(u) \rangle$ over all $u \in \mathcal{S}$
Let $\mathcal{C} \leftarrow$ SemanticClustering$(\mathcal{R}, \mathcal{S}, \kappa)$
Forall $seed \in \mathcal{S}$:
  Let $\mathcal{C}' \leftarrow \mathcal{C}$
  Update $\mathcal{C}'$ by removing locations in any partition with probability par_c
  Let semantic seed $semseed \leftarrow seed$
  Update $semseed$ by replacing locations r in it with $\mathcal{C}'_i$ where $r \in \mathcal{C}'_i$
  Update $semseed$ by removing the location $r = seed(t)$ from time t with probability par_l
  Update $semseed$ by merging locations that are located with time distance Δt with probability $par_m{}^{\Delta t}$
  Let $fake \leftarrow$ HMMDecode$(semseed, \langle \bar{p}, \bar{\pi} \rangle)$
  If PrivacyTest$(fake, seed, \mathcal{A}, \delta_s, \delta_i, \delta_d)$
    Let $\mathcal{F} \leftarrow \mathcal{F} \cup fake$
Return $\mathcal{F}$

Fig. 2: Trace synthesis algorithm. We present it simplified for the case with a single time period. If $\mathcal{C}$ is a semantic clustering, then $\mathcal{C}_i$ represents the set of locations belonging to the cluster i. The procedure HMMDecode is described in Section V-B.

we add some randomness to the semantic seed trace. In the transformation process of the seed trace into the semantic trace, we sub-sample locations from the semantic classes as opposed to using them all. We also remove each location in a cluster with probability par_c. The result is a new cluster C'. We allow locations of different classes to merge into each other around time instants where the user moves from one class to the other. We add a location from one time instant to another with a Δt gap with a geometric probability ${par_m}^{\Delta t}$.

B. Sampling a Trace from the Semantic Domain

Any random walk on the semantic seed trace that travels through the available locations at each time instant is a valid location trace that is semantically similar to the seed trace. However, the synthetic traces we want to generate also need to be geographically consistent with the general mobility of people in the considered area. We cast the problem of sampling such traces as a decoding problem in Hidden Markov Models (HMMs) [41]. The symbols are locations, the observables are the semantic classes (which are the set of semantically equivalent locations in the same class), and the transition probability matrix follows the aggregate mobility model.

We construct the aggregate mobility model by averaging over the mobility models of all traces in dataset $\mathcal{S}$, as well as giving a small probability to the possible movements between locations according to their distance and connectivity. More precisely, we compute the aggregate transition probability $\bar{p}_r^{r'}$ as $z_r^{-1} \cdot \sum_{u \in \mathcal{S}} p_r^{r'}(u) + \epsilon \cdot max(1, d(r, r'))^{-2}$, where ϵ is a small constant, and z_r is the normalizing factor. We compute the aggregate visiting probability $\bar{\pi}^r$ as the average of $\pi^r(u)$.[1]

By decoding the semantic trace into geographic traces, we generate traces that are plausible according to aggregate mobility models, i.e., there could be an individual who could have made that trace. Among existing HMM decoding algorithms, we use the Viterbi algorithm [51] that finds

$$\arg\max_{fake} \Pr\{fake | semseed(t), \langle \bar{p}, \bar{\pi} \rangle\}$$

assuming that $fake(t)$ can only choose from locations in $semseed(t)$. Finding the most likely fake trace is equivalent to finding the shortest path in an edge-weighted directed graph where each location at time instant t is linked to all locations at the subsequent time in the semantic seed trace.

By using this encoding technique, we make sure that the sampled trace is consistent with the generic mobility and has a significant probability of (geographically) being a real trace. However, the Viterbi algorithm produces a single trace (which is the most likely one). To generate multiple fake traces per seed, we add randomness to the trace reconstruction of Viterbi. We modify the Viterbi algorithm, which originally, at each step (time instant) selects the most probable location in the path; we add some randomness to the probabilities such that the algorithm does not deterministically select the most probable location. More precisely, we slightly perturb the probabilities in such a way that Viterbi selects randomly among a set of locations that are close in probability to the most probable location. We implement this idea by choosing a parameter par_v and multiplying all the probabilities of moving from one location to the next with a random number between 1 and par_v, which slightly randomizes our trace decoding.

Each generated trace is tested against our privacy test (Sections V-D and V-E). If it passes, we compute its likelihood based on the aggregate mobility model. At a later stage, we can randomly select fake traces to use based on this likelihood. Appendix A contains a brief discussion of the computational efficiency of the generation algorithm.

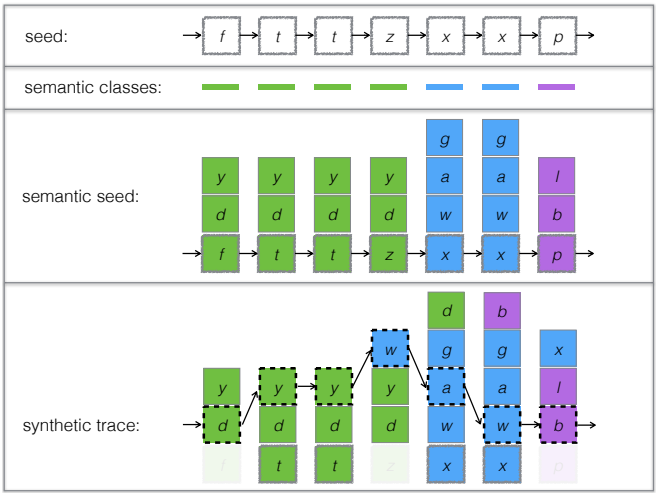


Fig. 3: A sketch of generating a fake trace from a seed. Each location is represented by an English letter in a box. The semantic class associated with each location is represented by a different color. The semantic seed trace is a trace that includes the locations in the seed along with other locations in the same cluster at each time instant. Here, locations are clustered as $\{y, d, f, t, z\}, \{g, a, w, x\}, \{l, b, p\}$. To generate a fake trace, we first probabilistically remove the seed location and probabilistically merge subsequent classes. In our example, f, z, p are removed, and w, d, b, x are merged into their neighboring visited clusters. We then run a decoder to generate a probable trace given the possibility of choosing from all available locations at each time instant. The fake trace, shown with connected dashed boxes, will be approved if it passes the privacy tests.

C. Threat Model

There are two types of privacy threats that we need to consider, which should not be confused with each other as they apply to different settings. The first threat is against the individuals whose traces are sampled and used as seeds in our algorithm. This is of more concern in the scenario of synthetic dataset release. In this scenario, the adversary knows that all traces are synthetic. Yet, he wants to extract geographic or semantic information about the seed traces, or find the identity of the individuals behind them. This is the threat that we are concerned about, in this section, in the process of generating synthetic traces. We define two privacy requirements to defeat

[1]We use the aggregate mobility model instead of that of a specific user to avoid constraining the reconstructed geographic trace to follow the user's profile too closely. For example, this means that the produced traces may visit locations which are never visited in the seed trace. This allows for greater utility because the input traces only cover a very small subset of the set of geographically meaningful traces.

this threat: *statistical dissimilarity* and *plausible deniability*. At the last step of synthesizing traces, we run a PrivacyTest() to enforce these two privacy requirements on each synthetic trace that is to be released.

The second type of attack is not perpetrated on the synthetic traces themselves. It is rather implemented against the queries received from LBS users that use the fake traces to hide their true locations. We do not need to address such threat here. However, we evaluate the effectiveness of our synthetic traces against location inference attacks in preserving location privacy of LBS users later in Section VII.

D. Privacy Requirement: Statistical Dissimilarity

A synthetic trace and its seed are statistically dependent, otherwise we cannot achieve utility. This is at the core of the privacy and utility tradeoff in any privacy-preserving sanitization algorithm. The goal is to guarantee some statistical privacy while preserving utility.

In our case, we guarantee a statistical dissimilarity between synthetic trace and its seed. To this end, two types of distance functions can be considered: (i) a distance between two probability distributions that model the synthetic trace and its seed. And, (ii) a distance between the two traces themselves.

As for the statistical distance (i), because there is a notion of (Euclidean) distance between locations that form a trace, we use Earth Mover's Distance between two probability distributions that represent the mobility models behind the two traces. This is exactly what we compute as geographic similarity metric. So, *for all seeds s and all synthetic traces f generated from s, we ensure that the statistical similarity between f and s is bounded by δ_s.*

$$\mathsf{sim}_{\mathsf{G}}(f, s) \leq \delta_s \tag{12}$$

As for the trace distance (ii) between f and s, we use the intersection between the two. In this case, *for all seeds s and all synthetic traces f generated from s, we ensure that the size of their intersection set is bounded by δ_i.*

$$|\mathsf{intersection}(f, s)| \leq \delta_i \tag{13}$$

We *reject* any synthetic trace that fails to satisfy either of the above conditions. Thus, we ensure a minimum statistical dissimilarity between a seed and its fake trace.

Intuitively, these tests prevent the leakage of privacy-sensitive locations. To understand why, consider Alice, a seed contributor, and the locations she visits daily, such as her home and workplace. In a released synthesized trace (from Alice's seed) these locations will not be visited. This is enforced by (12). But, what about atypical behaviors? Suppose Alice spends her morning at a women's health clinic (something out of the ordinary for her). (13) enforces that the women's health clinic will also not be visited in the released synthesized trace. However, locations visited that morning are likely to include health-related services locations (e.g., hospitals), since these may belong to the same semantic cluster. Could this information (i.e., that Alice visited a health-related location) be leaked? No, this is prevented by plausible deniability (Section V-E).

E. Privacy Requirement: Plausible Deniability

Enforcing a minimum statistical dissimilarity between a fake f and its seed s would limit the information leakage of f about s. In other words, this ensures a minimum error in reconstructing s by observing f. However, this is not enough to guarantee the location privacy of the individual associated with s due to the semantic similarity between s and f. Note that, due to utility requirements, our algorithm synthesizes traces f that are semantically very similar to s. Although, because of the randomness of our decoding algorithm, this semantic similarity varies, but it is mostly small. The threat is that an adversary, who has some background information about the individual associated with s, might be able to infer the inclusion of that individual's record in the seed dataset by observing f.[2]

The membership inclusion attack would work if s is by far the only real trace from which we could have generated f, i.e., when s is an *outlier*. To defeat such attacks, we introduce plausible deniability as a privacy requirement. To guarantee plausible deniability, we make use of the alternative real dataset $\mathcal{A}$ which is disjoint with the seed dataset $\mathcal{S}$. Concretely, the generated fakes (and their corresponding seeds) need to satisfy the following definition.

Definition 4. *A synthetic trace f (generated from seed $s \in \mathcal{S}$) satisfies* (k, δ)-plausible deniability *if there are at least $k \geq 1$ alternative traces $a \in \mathcal{A}$ such that:*

$$|\mathsf{sim}_{\mathsf{S}}(s, f) - \mathsf{sim}_{\mathsf{S}}(a, f)| \leq \delta \ . \tag{14}$$

In other words, for any fake f generated from seed s, we want to guarantee that we can find at least k alternative traces $a \in \mathcal{A}$ for which (14) holds for $\delta = \delta_d$. Specifically, the privacy test *rejects* all synthetic traces that do not satisfy (k, δ_d)-plausible deniability.

If condition (14) holds for a synthetic trace, then its seed is *not* the only real trace that could have generated it, and it is plausible that the same synthetic trace is generated from some other real traces (those that are even outside the seed dataset and have no contribution to generation of the synthetic trace). Thus, the inclusion of a particular trace in the seed dataset is plausibly deniable.

Intuitively, this safeguards the privacy of semantic outliers (i.e., seed contributors with atypical semantic behavior). To understand why, consider Alice, a seed contributor, who works the night shift, whereas most contributors work during the day. (14) enforces that Alice's synthetic trace can only be released if there exist k other semantically similar traces in $\mathcal{A}$ (i.e., only if at least k other alternative dataset contributors work during the night in a manner similar to Alice). Therefore, an adversary trying to run a membership inclusion attack will be thwarted even if he has the knowledge that Alice works during the night. Note that, as we select the alternatives from

[2]However, this does not imply that the adversary can accurately reconstruct s, especially if f satisfies the statistical dissimilarity requirements (12),(13).

outside the seed dataset, this holds even if Alice is the only seed dataset contributor who works during the night.

F. Discussion: Plausible Deniability as a New Privacy Notion

In this paper, we present plausible deniability as a new notion of privacy for data synthesis. In this section, we further discuss this notion and compare it with similar definitions in the literature. As said before, plausible deniability implies that a synthetic trace could have been generated from alternative location traces other than its own seed. This means that the information that is learned from observing a fake trace could have also been learned if the same fake trace was generated from other traces.

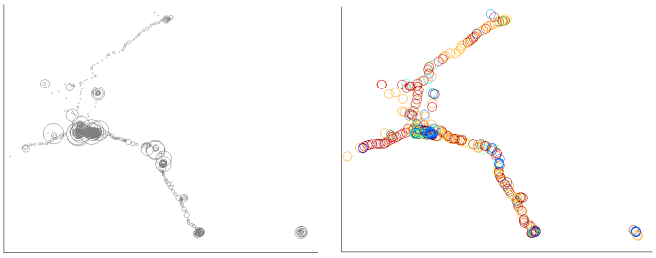
Note that we generate utility-preserving traces and release them only if they satisfy plausible-deniability privacy requirements. Other sanitization techniques in the literature, based on e.g., differential privacy [14] and crowd-blending privacy [16], enforce privacy in the process of sanitizing data. This makes it very challenging to design utility-preserving mechanisms under the constraints of privacy requirements.

To better understand the implications of plausible deniability on privacy, let us consider the cases where an adversary observes a synthetic trace generated from a seed trace that exhibits some rare characteristics, due to the particular lifestyle of the person who produced that trace. To avoid leaking about its seed, either (1) the synthetic trace must be semantically far enough from its seed so that it is equally close to alternative real traces, or else (2) there must already be alternative traces with similar rare characteristics. If neither of the two is true, the synthetic trace will not be released. This is similar to the notion of suppressing the sanitized data from outliers [16]. In fact, the overlap between the set of possible synthetic traces that can be generated from different real traces is the acceptable area for releasing synthetic data.

Plausible deniability, in spirit, is similar to some other notions of privacy. In crowd blending privacy [16] and its followup outlier privacy [31], the space from which data records are drawn is publicly split into partitions. Then, only the partitions that contain a minimum number of data points can produce sanitized data. The rest of data records, called outliers, need to perturbed with magnified noise, which would not lead to higher utility than simply suppressing them. The authors show that if crowd blending privacy is combined with subsampling of data records, it can achieve zero knowledge privacy [17] which is stronger that differential privacy [14].

Plausible deniability can also be guaranteed by differentially private mechanisms. Although it is possible, in theory, to generate fake traces in a differential private way, we do not know methods to do this efficiently due to the high-dimensionality of location traces. Specifically, this is practically infeasible given the existing mechanisms such as the Exponential Mechanism that require to assign a score to each possible trace given the input dataset.

Lastly, the plausible deniability privacy test, which requires each synthetic trace to be δ_d-indistinguishable from at least k alternative traces, should not be confused with k-anonymity.



(a) Visited Locations. The size of locations are proportional to their total visits.

(b) Visited locations colored according to their semantic clustering (20 clusters).

Fig. 4: 400 locations visited around Lausanne and nearby towns by the 30 users. Some users commute between two towns whereas the majority of them live and work in the same city of Lausanne (the area with higher concentration).

Unlike our notion of privacy, k-anonymity is a syntactic metric, achieved by suppressing or generalizing data, which does not prevent attribute disclosure. It has also been shown to be severely vulnerable to inference attacks when used to protect location privacy [46].

VI. Evaluation Setup

In this section, we run our algorithms on a set of real location traces and evaluate the resulting utility and privacy in two scenarios: sharing locations with LBS, and releasing synthetic location datasets.

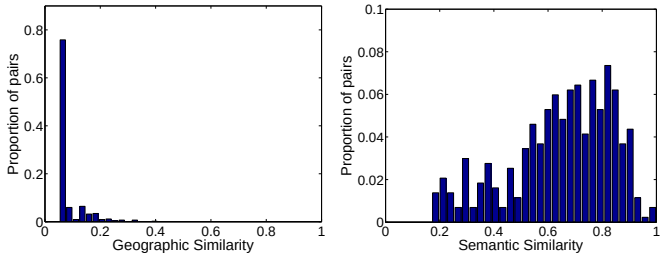
A. Dataset

The dataset we use for the evaluation is collected through the Nokia Lausanne Data Collection Campaign (see [25]). We prepare the dataset for our needs in two phases, filling gaps in the traces and discretizing the time and location.

The raw dataset contains combination of events of three types: GPS coordinates, WLAN and GSM identifiers. We construct valid traces (out of partial traces) by filling gaps. We interpolate along the path of consecutive GPS points, using the WLAN and GSM information.

We then extract two days of traces for each user, such that each trace (of one day) contains a sequence of 72 locations (i.e., one location is reported every 20 minutes). Some locations are visited very rarely only by very few users. Thus, we reduce the number of locations from 1491 to 400 by clustering close-by locations together. We use a hierarchical clustering algorithm for this purpose, and place the locations that are geographically close or have very few visits in one cluster. The geographical distribution of visits of all users over the locations in the considered area is shown in Figure 4(a).

From all traces, we then sub-sample 30 user traces. The 1st day of traces for these users is used as seed dataset $\mathcal{S}$, whereas the 2nd day of traces will be used as baseline (testing set) during the evaluation. Using the seed dataset, we compute the mobility profiles of all 30 users, and then the semantic location graph by calculating a similarity score for each pair of locations, averaged across all users. After clustering this semantic location graph, we obtained 20 location clusters. We choose this number of clusters as it provides optimal clustering



(a) (b)

Fig. 5: Normalized histogram of (a) the geographic similarity and (b) semantic similarity between all distinct pairs of 30 users in the seed dataset. (a) Mobility models of different individuals is geographically very specific to themselves, i.e., they are unique. This is well reflected in the skewed distribution of geographic similarity towards very small values. (b) As hypothesized in this paper, majority of individuals have high semantic similarities between their mobility models.

i.e., it maximizes the ratio of inter-cluster similarity over intra-cluster similarity. This clustering is illustrated in Figure 4(b), where each location is drawn with the color of the cluster it belongs to. The figure allows us to distinguish some patterns, for example locations at the center of cities are mostly in blue, while many locations representing roads and highways are colored in red. Also notice that the semantic clustering does not seem to depend on the geographical distance of locations.

To illustrate our geographic and semantic similarity metrics, we compute those metrics pairwise over all 30 users.[3] The result is shown in Figures 5. The first histogram shows that the 30 users are not strongly geographically similar to each other, except for a few pairs of users. This is expected given the range of locations they explore overall, as seen in Figure 4(a). On the other hand, the distribution of the semantic similarity across all distinct pairs of users has a larger variance, and a large number of users are highly similar.

B. Tool: Synthetic Trace Generator

We build a tool [6] to generate fake traces on top of the open-source Location Privacy Meter (LPM) [46]. To exploit LPM's modularity we split our algorithm into modules. To implement the time-dependent sub-sampling of clusters and merging around transitions, and the transformation of users' actual traces into semantic traces, we use the location obfuscation mechanism feature of the tool. The reconstruction of geographically valid synthetic traces from the semantic traces is done using the Viterbi algorithm. To cluster the semantic location graph, we employ the CLUTO toolkit [53].

C. Simulation Setup

Recall that from the input dataset, we sub-sampled 30 user traces (day 1) that we use for the seed dataset $\mathcal{S}$. As for the parameters of the GenerateFake() algorithm, we set the location-removal probability par_c to 0.25, and we set the location merging probability par_m to 0.75. We set the probability par_l of removing the true location visited in the seed to 1.0. We set the randomization multiplication factor for Viterbi randomization par_v to 4.

We set very tight values for the privacy parameters. Specifically, we set δ_i, the maximum intersection between fake and seed, to 0. So, we do not tolerate any intersection between fake and seed. We set the geographic similarity threshold δ_s to 0.1, and the differential semantic similarity threshold δ_d also to 0.1. See Figure 5 to see comparatively how restrictive these thresholds are. Last, we set k the number of required alternatives to pass the plausible deniability to 1.

For each of the 30 seed traces, we generated about 500 fake traces. We then select and use these traces according to the scenario evaluated. For example, for the LBS scenario, we sampled traces (for each user) according to the synthetic traces likelihoods (see Section V), out of the pool of traces that passed the privacy test.

D. Evaluation Metrics

In the following two sections we evaluate the use of synthetic traces in two popular scenarios: using fake locations along with real locations when accessing location-based services (Section VII), and releasing synthetic location datasets to be used for various geo-data analysis tasks (Section VIII). In both scenarios, we evaluate our fake traces with respect to two metrics: *privacy* and *utility*. Our privacy guarantees apply to both scenarios. However, there are some differences in terms of the adversary model between different scenarios. Therefore, there are additional considerations regarding the privacy of users in location-based services, e.g., their privacy against inference attacks, that we discuss in the corresponding section. The utility metric is also dependent on the application (scenario), hence is measured differently in each case.

Note that the generative power of our model (and similarly any statistical or machine learning model) depends on the available (training) data. Yet, similar to a machine learning algorithm, we do not require a "minimum" number of data records to start working. However, the quality of the input dataset does not impact data privacy, as privacy is guaranteed in the phase when we are *generating* traces (and not in the phase when model is trained), by running our privacy tests. So, the output is privacy preserving regardless of the size of the input and quality of the model.

VII. Evaluation: Location-based Services

In this section, we consider the use of fakes in accessing location-based services (LBS). Specifically, we compare our proposed technique with all existing fake generation methods. Concretely, we evaluate the utility and privacy according to well-established metrics for this scenario. In particular, we measure how well our fakes perform against state-of-the-art inference attacks. Note that the fakes used here have already passed our privacy tests with tight constraints, but as explained in Section V-C, we still need to test their effectiveness in protecting privacy of LBS users against inference attacks.

[3] We exclude the geometric/semantic similarity of a user with herself, as it's 1.0.

A. Setup

In this setting, a user shares her current location with a location-based service. The service provider, in return, provides contextual information about the shared locations (e.g., list of nearby restaurants, current traffic information on the road). The user makes such queries over time whenever she wishes to obtain contextual information.

In order to protect her location privacy, i.e., hiding her location at the time of access to the LBS and also preventing the inference of the full trajectory, the user's device sends a number of fake locations along with her true location. For example, if two fake locations are used, then every time the user makes a query, the device sends locations x, y, z to the service provider. Out of $\{x, y, z\}$, one location is the user's actual (i.e., true) location and the other two are fakes. The service provider does not know which of x, y, z is the true location, but may be able to filter out fake locations over time (i.e., over multiple queries over time), if the fake locations are not believable (i.e., plausible). This is why it is crucial to use synthetic *traces* as opposed to independent fake *locations*.

The fake locations are obtained as followed. First, we generate a collection of synthetic traces. The users can select from these traces and store them in their devices. Then, when a user makes an LBS query, say at time t, she picks the i^{th} fake location (reported to the LBS) as the location which is visited at time t in fake trace i. Note that the processes of generating and using the synthetic traces are independent.

We emphasize that all existing fake location generation methods (i.e., [10], [23], [24], [26], [45], [49], [56]) work this way; but the techniques differ in how the fake locations are generated. Existing fake locations generation methods can be classified into four categories, which are the following.

Uniform IID ([45]): Generate each fake location independently and identically distributed from the uniform probability distribution. So, the fake trace is a sequence of uncorrelated fake locations.

Aggregate Mobility IID ([45]): Generate each fake location independently and identically distributed from the aggregate mobility probability distribution $\bar{\pi}$. Again, the fake trace is a sequence of uncorrelated fake locations.

Random Walk on Aggregate Mobility ([10], [24], [26], [56]): Generate a fake trace by doing a random walk on the set of locations following the probability distribution $\bar{p}$.

Random Walk on User's Mobility ([23], [49][4]): Do a random walk on the set of locations following the probability distribution $p(u)$ to generate a fake trace.

For Uniform IID and Aggregate Mobility IID, we evaluate exactly the method described in [45]. For the other two we evaluate a representative method in each case. In addition to this, we evaluate our proposed technique, which only differs from these in that the fake traces are generated using the method described in Section V. In all cases when a user makes use of a location-based service, both a query for her real location *and* queries for the fake locations are sent to the LBS. Because of this, the user's device must, upon receiving the responses from the service provider, filter out the information which are not related to her true query.

Using a Uniform IID mechanism may seem too simplistic and unfair, but we point out that the related work [45] evaluates this technique so we provide it as a point of comparison.

B. Privacy Metric

The adversary (e.g., the service provider) who observes the LBS queries made by the user's device wants to find the true sequence of locations visited by the user. To do this, the adversary runs an inference attack which (if successful) results in filtering out the fake locations. For this, he makes use of the aggregate mobility model $\langle \bar{p}, \bar{\pi} \rangle$ and uses state-of-the art location inference attack [46]. The attack is a localization attack which consists in finding the user's (true) location at each time, given the observation (i.e., the sequence of locations queried to the LBS). This is a well-known inference problem for Hidden Markov models which can be solved efficiently using dynamic programming.

The metric to quantify the privacy is the *probability of error* of inference attack on guessing the correct location. This is the metric predominantly used in the literature, in works such as [45], [46]. To put it simply, this metric consists in calculating the fraction of true locations that are missed by the adversary. For example, if the user queries LBS on three different occasions, but the adversary only correctly infers the true location once (i.e., the inference attack correctly filters out the fake locations) then the user's location privacy is $2/3$.

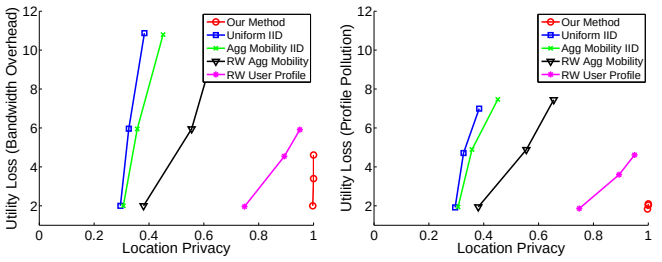
C. Utility Metric

With all synthetic generation techniques (i.e., ours and [10], [23], [24], [26], [45], [49], [56]), the user's real location will always be among the locations queried to the LBS. Therefore, as identified by related work, there is no utility loss in terms of quality of service degradation. That is, the user will always obtain an accurate response to her query (after filtering out responses corresponding to fake location queries).

Therefore, we measure the utility loss as the *bandwidth overhead*. This is the metric used in the literature on fake generation techniques (e.g., [56], [24], [23]). The bandwidth overhead is calculated as the number of locations (i.e., real + fakes) sent to the LBS provider for each user query.

Beyond traditional location-based services, some service providers (e.g., Google Now) profile the user's interest over time based on the type of locations she visits. This is to provide recommendations or reminders. In such cases, queries that are sent to the server can "pollute" the user's profile, hence reduce the predictability power of the service provider to provide useful recommendations. To further evaluate utility for such location-based recommender services, we calculate the number of (distinct) semantic clusters among the locations sent by the user at each time. We call this metric: *profile pollution*.

[4] [23], [49] make fakes dependent on the user's location over time (used to establish the position of dummies). We make this probabilistic and so assume usage of the user's mobility profile instead. This leads to overestimating the privacy gain of the original algorithm.



(a) (b)

Fig. 6: Location privacy versus utility loss for different fake generating algorithms. The privacy is measured as probability of error of adversary in guessing the correct location of users ([45], [46]). We plot the median location privacy across all LBS users. A user makes, on average, an LBS query every 40 minutes. We evaluate the use of $1, 5, 10$ fake traces, hence three dots for each algorithm. (We repeated the experiment 20 times and took the average: 4 times with a different selection of fake traces, and for each of such selection, 5 times to eliminate the randomness.) The utility loss is (a) the bandwidth overhead ([56], [24]), i.e., number of distinct locations that are sent to the server; and (b) the profile pollution, i.e., the number of distinct semantic classes exposed for each LBS access.

D. Results

Figure 6 shows the tradeoff between location privacy and utility for various methods of generating fake traces. We evaluate the utility loss in terms of two metrics: bandwidth overhead (Figure 6a) which is predominantly used in the literature, and also the profile pollution (Figure 6b). We evaluate the privacy for three different number of fake traces: $1, 5, 10$. Although the number of fake traces are the same, across different algorithms, the average number of distinct locations sent to the LBS is not the same. This is because of the potential overlap between fake traces available to the user. Methods such as *Uniform IID*, *Agg Mobility IID*, and *RW Agg Mobility* have a high randomness in selecting fake traces from all possible locations. Thus, the chance of overlap is small. Our method and the *RW User Profile* method have both lower bandwidth overhead.

Results show that our method clearly outperforms all the existing techniques, especially the random strategies. For the case of *RW User Profile* method, the privacy level against the tracking attack gets closer to what we achieve (which is almost maximum), due to the fact that the fake traces generated by *RW User Profile* are semantically very similar to the user's locations, and hence creates high confusion, hence error, for the adversary. However, it is very important to note that the *RW User Profile* is never a privacy-preserving fake injection method as the adversary can easily de-anonymize the user, no matter if he makes mistakes on exactly tracking the user at each access time (as shown here).

Overall, the plot shows that our method is the strongest fake generating algorithm. Note that the absolute privacy levels changes if the adversary knowledge changes. But, what we are interested in is the relative gain of our method to others.

VIII. Evaluation: Synthetic Data Release

In this scenario, the synthetic traces form a location dataset that is meant to be used for various geo-data analysis tasks *in place of* real location traces. We emphasize that the seed traces and the alternative dataset are *not* released. Only the generated synthetic traces are supposed to be released.

A. Setup

We generate a large number of fake traces out of which we ultimately select: 10 datasets each containing 30 traces. This is done in order to have each fake dataset of the same size and format as the seed dataset, so that we can compare the suitability of using one of those fake dataset instead of the seed dataset for various geo-data analysis tasks.

B. Privacy

The location privacy of those individuals who contributed to the seed dataset is already guaranteed by our use of the privacy test. However, we must make sure that we are able to generate traces which pass the privacy test with acceptably tight constraints. Therefore, this is what we evaluate here. Out of all fake traces generated from our 30 seed traces, on average 80% of them could pass the geographic and intersection privacy tests with tight constraints ($\delta_i = 0$, and $\delta_s = 0.1$), so it is not difficult to synthesize traces that satisfy such privacy guarantees. Regarding the plausible deniability part of the test, the question is whether we are able to find enough real traces in alternative dataset $\mathcal{A}$.

In Figure 7, we show the difference between semantic similarity of a synthetic trace and its seed with the semantic similarity of the same synthetic trace and any location trace in our alternative dataset. The histogram shows that the majority of fake traces have very low distinguishability to alternative traces, in the semantic domain. This is due to the high semantic similarity between real traces (Figure 5b). Therefore, we conclude that it is not difficult to find potential alternative traces for a synthetic trace. Recall that we set δ_d to 0.1 to obtain a high level of plausible deniability.

C. Utility

Because we release a set of synthetic traces to be used instead of a real location dataset, to evaluate utility we must take into account how the released traces are to be used. Specifically, we must determine to what extent the key features and statistics, which are relevant for the considered applications, are preserved. Clearly, we cannot expect all statistics (of real traces) to be preserved (in the synthetic dataset) since some may conflict our privacy goal. Specifically, certain geographic features are expected not to be preserved, e.g., if a location is primarily visited by a single user in a seed, it is unlikely that that location is visited with similar frequency in a synthetic trace. This is because if such a synthetic trace were generated from that seed, the privacy test would reject it. An example of property that we do not preserve is the relationship in the mobility of input traces, e.g., individuals commuting to work together. Indeed, if two individuals carpool to work, the corresponding synthetic traces will not exhibit analogous co-traveling behavior (because each synthetic trace is generated

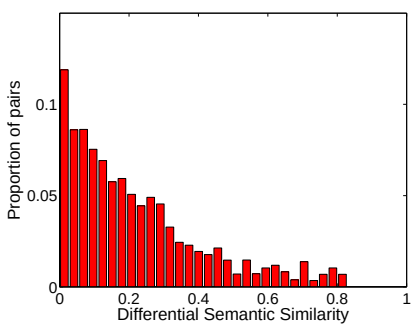


Fig. 7: Histogram of the differential semantic similarity between fake and real traces. It presents the distribution of the absolute difference $|\mathsf{sim}_S(s, f) - \mathsf{sim}_S(s', f)|$, for all pairs of f (plus its seed s) and s'.

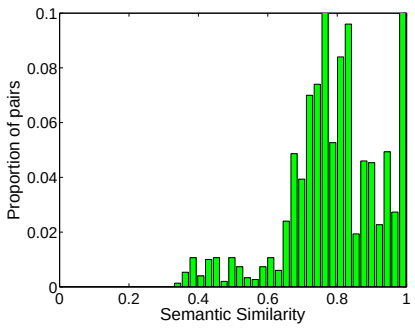


Fig. 8: Normalized histogram of the semantic similarity of all distinct pairs of: each of the 30 real traces, along with their associated fake traces.

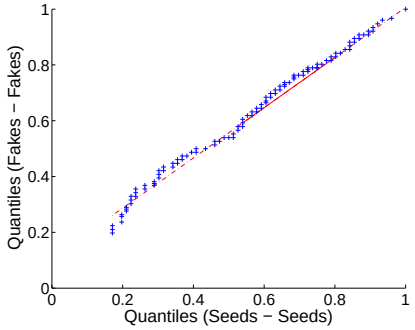


Fig. 9: Q-Q plot for comparing two distributions: semantic similarity among all real seed traces, and semantic similarity among all fake traces. The plot shows a very strong correlation between two distributions.

independently). However, their common semantic features are preserved.

That said, from the literature, we identified the following prominent geo-data analysis tasks to evaluate utility of synthetic traces.

(1) Points of Interests (PoIs) extraction. The goal is to discover locations that are frequently visited and are prominently of interest to the public. PoIs can be used to provide travel recommendations. In particular, [59] proposes techniques to mine the top n interesting locations in a given region. A key feature to preserve is the distribution of visits among locations, specifically the most visited (i.e., popular) locations.

(2) Semantic annotation / labeling of locations. The goal is to automatically assign labels to locations according to their semantics (e.g., restaurant, bar, shopping mall). For example, [55] proposes an SVM classifier to assign multiple labels to location-based social network check-ins. In contrast, [13] proposes to do automatic labeling of locations into 10 semantic categories using smartphone recorded GPS, WiFi, and cell-tower data. In all cases, the distribution of visitors (and unique visitors) per location are key features of the input data. In addition, [13], [55] use users' temporal behavioral data, such as the amount of time a user spends in a location.

(3) Map inference. [29] evaluates the two main approaches to infer road maps from a large scale GPS traces: using the sample coordinates themselves, or using the transitions between samples. A related task is the discovery of semantic regions in a city [57]. In both cases, key features of the input data include the distribution of visited locations, and transitions, particularly the popular ones.

(4) Modeling human mobility. [28] proposes to learn a multi-layer spatial density model from social network check-ins. In this case, temporal features of location data are largely overlooked. Rather the focus is on features such as the spatial location distribution in aggregate and at individual level.

(5) Determining optimal locations for retail stores. The goal is to find ideal geographic placement for a retail store, or a new business. In particular, [22] proposes to mine online location-based services to evaluate the retail quality of a geographic area. Specifically, the focus is on a combination of mobility features such as popularity of an area, and semantic features such as visits to semantically similar venues (e.g., coffee shops of the same franchise) or transitions between venues.

Based on the input features that those geo-data analysis tasks require, we identified six statistics that need to be preserved to guarantee that such tasks can reasonably be performed on a set of synthetic traces *instead of* a real dataset. In order to experimentally evaluate to what extent these statistics are preserved, we must use appropriate baselines. We use the value of the statistic on the testing (day 2) dataset, which consists of location traces of the same users as the seed (day 1) dataset, as the baseline. When appropriate, we also use uniformly random location traces as a baseline.

The corresponding useful features are the following.

(a) Distribution of the number of visits. Tasks such as (2) and (3) exploit the fact that some locations are more frequently visited than others. In fact, [13] explicitly mention "how often places are visited" as a major feature.

In order to evaluate this, we do the following. For each dataset, we compute the spatial allocation, i.e., for each location (from least to most popular, for that dataset), we calculate the number of visits spent in that location across all traces in the dataset. We then normalize this quantity to obtain a probability distribution over locations (sorted by popularity), i.e., for each location we have the probability of a random visit to that location. From these distributions, we compute the KL-divergence of the real (seed, i.e., day 1) dataset to each of our synthetic datasets, and to a variety of baselines.[5] The KL-divergence is a natural way to compare distributions: it returns a non-negative real number, where a larger value means a greater distance between the two distributions. Note that some related work such as [9] has used the relative error of counting

[5]We set zero probabilities to 0.1, before normalizing, for the sake of computing KL-divergence (that requires nonzero probabilities). This is required because there may be locations which are visited in the synthetics but not in the real traces, or vice-versa.

	Real	Synthetic	Uniform	Single
KL-divergence	0.037	0.384 ± 0.043	1.191	4.666
Relative error [9]	0.144	0.370 ± 0.010	1.621	0.542

TABLE II: KL-divergence and relative error of the location visiting probabilities of the real (seed, day 1) datasets against the 10 fake datasets, and various baselines. "Real" is the testing (day 2) portion of the real dataset (see Section VI-C); "Uniform" is the uniform distribution over all locations; and "Single" is the distribution where all users always visit the same location.

queries as a metric instead of the KL-divergence. Therefore, we additionally calculate the relative error by interpreting the number of visits to each location as the answer to a counting query. That is, if the number of visits to location x is n_1 for dataset 1 and n_2 for dataset 2, then we calculate the relative error as $\frac{|n_1-n_2|}{\max(n_1,0.001\cdot N)}$, where N is the total number of visits to any location (the same for all the datasets) [9]. We report the average relative error over all locations. The results of both metrics are shown in Table II. The results suggest that a lot of information is preserved in this case: while the error for the fakes is greater than that for the real (testing, day 2) dataset, the error is significantly lower than the other baselines.

(b) Distribution of number of visits for top 50 ***locations.*** For most tasks, features of the most popular locations (i.e., the most frequently visited locations) are the most important ones to preserve. In particular, this is consistent with the results provided in [13] for automatic labeling.

To evaluate this, we use the same procedure as for feature (a), except we only consider the top 50 locations, and plot the distribution instead of calculating the KL-divergence. Figure 10 shows the results for this case, which plots a histogram where the distributions for different datasets are overlayed (with some transparency). The error (of the synthetic dataset) for this case (i.e., top 50) is significantly lower than that for the entire distribution. This strongly indicates that the information about the popular locations (i.e., the most important ones) is largely preserved.

(c) Top n ***coverage of locations.*** For tasks such as (1), (3), and (5), it may not be sufficient to ensure that the distribution of visits is preserved. Indeed, it may be required to ensure that if a location is in the top n most frequently visited locations in the real dataset, it is also in the top n most frequently visited locations in the released (synthetic) dataset.

Therefore, we measure across two datasets (e.g., one real and one synthetic), how many locations in their respective top-n they have in common, for various values of n.

Specifically, we take the n most frequently visited locations of the real (seed, day 1) dataset. For each of the other datasets, we then compute how many top n locations (from the seed dataset) are also in the top n most frequently visited locations of that dataset. For the synthetic datasets and the uniform baseline, we report the relative coverage as the ratio of the coverage of that dataset and of the testing (day 2) dataset. That is, if the coverage of the real (testing, day 2) dataset is y (of the top n locs of the seed dataset), and the average coverage of the fake datasets is x, then we report the relative coverage as $\min(\frac{x}{y}, 1.0)$. The results are shown in Figure 11. The relative coverage of the synthetic traces is typically in the 61% to 100% range, whereas for the uniform baseline it is in the 11% to 24% range, indicating a high-level of preservation.

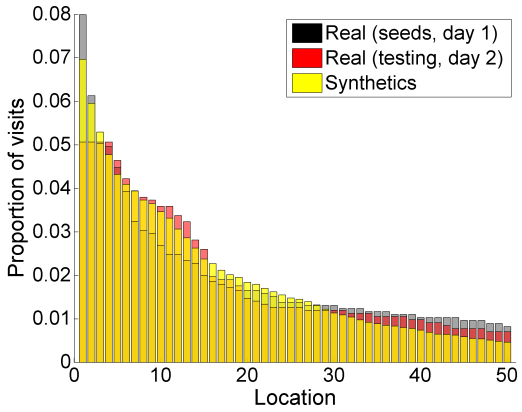


Fig. 10: Distribution of visiting probability for top-50 locations in the real (seeds) datasets against the synthetic datasets. We overlay (with some transparency) the histograms of the three datasets (i.e., seeds day 1, in black; real day 2, in red; synthetics, in yellow). The difference in distribution between two datasets is the area where the two corresponding histograms' bars do not overlap. For example, the lightest yellow region is where the synthetics' histogram is non-overlapping with the other two histograms; the darker yellows are areas where the synthetics' histogram overlaps with one (or both) of the real datasets' histograms. The majority of the colored area is a region where all three datasets overlap, indicating that the synthetics highly preserve the distribution of the top-50 most popular locations.

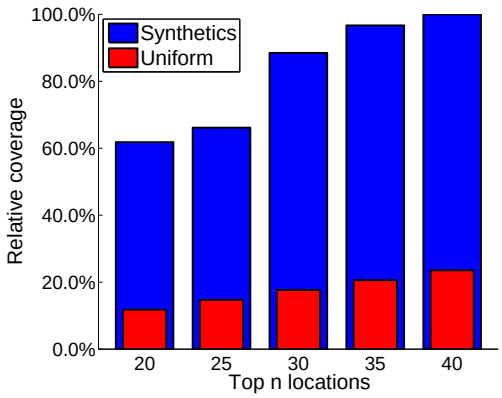


Fig. 11: Relative coverage of top n (most frequently visited) locations. The coverage is reported relative to the real (testing, day 2) dataset. Uniform visiting of all locations (400 in total) is used as comparative baseline.

(d) Users' time allocation. For semantic labeling (2) and other tasks, the users' temporal behavior cannot be ignored. Indeed both [13], [55] use the amount of the time spent per location for each user as a major feature.

In order to evaluate this, we do the following: for each dataset and each user, we calculate the time spent at each location, among the locations visited. That is, we calculate, for the three most popular locations of that user, what proportion of the time is spent in each. We perform this calculation across all 30 users and normalize the result. We compare this

	Real (day 2)	Synthetics	Uniform	Random
1st	0.0189	0.0125 ± 0.0022	0.1652	0.6794
2nd	0.0026	0.0092 ± 0.0031	0.0778	0.5360
3rd	0.0114	0.0089 ± 0.0036	0.0779	0.5092

TABLE III: KL-divergence of the users' time allocation distribution among the three most popular locations (of each user) of the real (seed, day 1) datasets against synthetic datasets, and baselines.

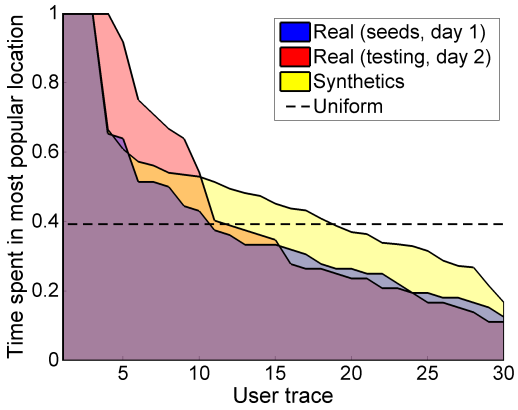


Fig. 12: Distribution of the proportion of time spent in the most popular location (of each user) of the real datasets against synthetic datasets. The information is presented as an area plot, where the distribution for each dataset is plotted as surface of a different color (i.e., seeds day 1, in blue; real day 2, in red; synthetics, in yellow). The areas are overlayed on top of one another. Therefore, the distance between the distribution of two datasets is represented by their non-overlapping area. For example, the yellow and orange regions represent areas where the synthetics' distribution is either non-overlapping (yellow) or overlaps with the real day 2 dataset's distribution, but not with seeds (day 1) dataset. The majority of the colored area is a region where the real and synthetics distributions overlap (i.e., purple region). This indicates a high-level of preservation.

distribution for the real and synthetic datasets. Table III shows the KL-divergence of the real (seed, day 1) dataset to the synthetic datasets and baselines: real (testing, day 2) dataset; uniform time allocation (each user spends $1/k$ proportion of time at each of the k locations); random time allocation (each user spends a uniformly random proportion of time at the location). To visualize those results further, Figure 12 shows the distribution across all 30 users (for each dataset) for the most popular location (only). The statistic is highly preserved in the synthetic traces; sometimes the synthetics' distribution is closer to that of the real (seed, day 1) dataset, than the distribution of the real (testing, day 2) dataset is.

(e) Spatiotemporal mobility features. When constructing mobility models from location data (4), the overall geographic and temporal behavior of users' mobility is used.

To evaluate this, we compare the basic mobility statistics obtained from the real and synthetic datasets. We compute the aggregate mobility model for each synthetic dataset, and compare its geographic similarity with the real (seeds, day 1) dataset. More precisely, for a synthetic dataset $\mathcal{F}$, we compute $\langle \bar{p}_{\mathcal{F}}, \bar{\pi}_{\mathcal{F}} \rangle$ and compute its similarity to $\langle \bar{p}, \bar{\pi} \rangle$. The statistical similarity of $\bar{p}_{\mathcal{F}}$ with $\bar{p}$ over all synthetic datasets is

$$[0.8061\,(average), \quad 0.8073\,(median), \quad 0.0060\,(std)],$$

and the results for the statistical similarity of $\bar{\pi}_{\mathcal{F}}$ with $\bar{\pi}$ is

$$[0.7856\,(average), \quad 0.7867\,(median), \quad 0.0152\,(std)].$$

Both these results show a strong correlation between average/aggregate mobility information of real and fake datasets.

(f) Semantic mobility features. In contrast to other applications, identifying areas for new businesses, i.e., task (5) explicitly takes into account semantic features of the input location data. Specifically, it takes into account visits to semantically similar venues and transitions between different types of venues. Consequently, it is meaningful to measure the extent to which semantic features of a real dataset are preserved in a released fake dataset.

To evaluate this, we proceed in two steps. We first compute the semantic similarity of each synthetic trace with its own seed trace to check if the semantic features of the original traces are indeed preserved. Figure 8 illustrates the distribution of this value over all fake traces. Clearly, the distribution is biased towards higher similarity values. So, the fake traces considerably preserve the semantic features of the real traces.

In the second step, we look at whether the set of synthetic traces preserves the inner similarity between the set of traces. In Figure 9, we present the correlation between two distributions: semantic similarity among real traces, and semantic similarity among synthetic traces. The Q-Q plot shows a significant correlation between these two distributions; they are strongly linearly related. This reflects that in addition to maintaining the information about each seed, we also preserve the statistical relation among the traces.

Overall, the statistics we have identified are largely preserved in the synthetic datasets. Thus, we conclude that our technique is suitable for the aforementioned geo-data analysis tasks and those that rely primarily on similar features.

IX. Conclusions

This is the first paper to systematically generate plausible synthetic location traces based on quantitative metrics. We propose statistical dissimilarity and plausible deniability as privacy requirements for synthesizing location traces. By enforcing these requirements, synthetic traces would not leak information about real traces from which they are generated more than what they have in common with any random real trace. Through extensive privacy and utility evaluations, we show the application of our mechanism in two mainstream scenarios: protecting the location privacy of users in LBSs, and geo-data analysis on synthetic location data. Our synthesized traces can be of extreme help in protecting location privacy of LBS users with very low utility cost. We show that inference attacks cannot identify the true location of mobile users if our fake traces are used as protection. We also quantitatively show that our method is superior to all existing methods of generating fake traces. Our synthetic traces also preserve useful features of real traces and can be useful in five popular geo-data analysis tasks.

REFERENCES

[1] G. Acs and C. Castelluccia, "A case study: Privacy preserving release of spatio-temporal density in paris," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. New York, NY, USA: ACM, 2014, pp. 1679–1688. [Online]. Available: http://doi.acm.org/10.1145/2623330.2623361

[2] G. Acs, C. Castelluccia, and R. Chen, "Differentially private histogram publishing through lossy compression," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 1–10.

[3] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 901–914.

[4] C. A. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *Dependable and Secure Computing, IEEE Transactions on*, vol. 8, no. 1, pp. 13–27, 2011.

[5] O. Berthold and H. Langos, "Dummy traffic against long term intersection attacks," in *Privacy Enhancing Technologies*. Springer, 2003, pp. 110–128.

[6] V. Bindschaedler and R. Shokri, "Tool: Plausible privacy-preserving location trace generator." [Online]. Available: https://vbinds.ch/projects/sglt

[7] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 251–262.

[8] S. Brooks and B. Morgan, "Optimization using simulated annealing," *The Statistician*, pp. 241–257, 1995.

[9] R. Chen, G. Acs, and C. Castelluccia, "Differentially private sequential data publication via variable-length n-grams," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 638–649.

[10] R. Chow and P. Golle, "Faking contextual data for fun, profit, and privacy," in *WPES '09: Proceedings of the 8th ACM workshop on Privacy in the electronic society*. New York, NY, USA: ACM, 2009, pp. 105–108.

[11] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 1994.

[12] C. Diaz and B. Preneel, "Taxonomy of mixes and dummy traffic," in *Information Security Management, Education and Privacy*. Springer, 2004, pp. 217–232.

[13] T. M. T. Do and D. Gatica-Perez, "The places of our lives: Visiting patterns and automatic labeling from longitudinal smartphone data," *Mobile Computing, IEEE Transactions on*, vol. 13, no. 3, pp. 638–648, 2014.

[14] C. Dwork, "Differential privacy," in *33rd International Colloquium on Automata, Languages and Programming, ICALP 2006*, ser. Lecture Notes in Computer Science, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052. Springer, 2006, pp. 1–12.

[15] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM, 2009, pp. 371–380.

[16] J. Gehrke, M. Hay, E. Lui, and R. Pass, "Crowd-blending privacy," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 479–496.

[17] J. Gehrke, E. Lui, and R. Pass, "Towards privacy for social networks: A zero-knowledge based definition of privacy," in *Theory of Cryptography*. Springer, 2011, pp. 432–449.

[18] A. Gervais, R. Shokri, A. Singla, S. Capkun, and V. Lenders, "Quantifying web-search privacy," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 966–977.

[19] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving privacy in gps traces via uncertainty-aware path cloaking," in *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2007, pp. 161–171.

[20] D. C. Howe and H. Nissenbaum, "TrackMeNot: Resisting surveillance in web search," *Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society*, vol. 23, pp. 417–436, 2009.

[21] A. Juels and R. L. Rivest, "Honeywords: Making password-cracking detectable," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 145–160.

[22] D. Karamshuk, A. Noulas, S. Scellato, V. Nicosia, and C. Mascolo, "Geo-spotting: Mining online location-based services for optimal retail store placement," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 793–801.

[23] R. Kato, M. Iwata, T. Hara, A. Suzuki, X. Xie, Y. Arase, and S. Nishio, "A dummy-based anonymization method based on user trajectory with pauses," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 2012, pp. 249–258.

[24] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Pervasive Services, 2005. ICPS '05. Proceedings. International Conference on*, July 2005, pp. 88–97.

[25] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila, "Towards rich mobile phone datasets: Lausanne data collection campaign," *Proc. ICPS, Berlin*, 2010.

[26] J. Krumm, "Realistic driving trips for location privacy," in *Pervasive '09: Proceedings of the 7th International Conference on Pervasive Computing*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 25–41.

[27] E. Levina and P. Bickel, "The Earth Mover's distance is the Mallows distance: some insights from statistics," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, 2001, pp. 251 –256 vol.2.

[28] M. Lichman and P. Smyth, "Modeling human location data with mixtures of kernel densities," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 35–44.

[29] X. Liu, J. Biagioni, J. Eriksson, Y. Wang, G. Forman, and Y. Zhu, "Mining large-scale, sparse gps traces for map inference: comparison of approaches," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 669–677.

[30] H. Lu, C. S. Jensen, and M. L. Yiu, "Pad: privacy-area aware, dummy-based location privacy in mobile services," in *MobiDE '08: Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*. New York, NY, USA: ACM, 2008, pp. 16–23.

[31] E. Lui and R. Pass, "Outlier privacy," in *Theory of Cryptography*. Springer, 2015, pp. 277–305.

[32] C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao, "Privacy vulnerability of published anonymous mobility traces," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, ser. MobiCom '10. New York, NY, USA: ACM, 2010, pp. 185–196. [Online]. Available: http://doi.acm.org/10.1145/1859995.1860017

[33] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber, "Privacy: Theory meets practice on the map," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. IEEE, 2008, pp. 277–286.

[34] D. J. MacKay, *Information theory, inference, and learning algorithms*. Citeseer, 2003, vol. 7.

[35] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[36] D. J. Mir, S. Isaacman, R. Caceres, M. Martonosi, and R. N. Wright, "Dp-where: Differentially private modeling of human mobility," in *Big Data, 2013 IEEE International Conference on*, Oct 2013, pp. 580–588.

[37] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial & Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[38] P. M. Pardalos, H. Wolkowicz *et al.*, *Quadratic Assignment and Related Problems: DIMACS Workshop, May 20-21, 1993*. American Mathematical Soc., 1994, vol. 16.

[39] A. Pingley, N. Zhang, X. Fu, H.-A. Choi, S. Subramaniam, and W. Zhao, "Protection of query privacy for continuous location based services," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1710–1718.

[40] D. Proserpio, S. Goldberg, and F. McSherry, "Calibrating data to sensitivity in private data analysis," *Proceedings of the VLDB Endowment*, vol. 7, no. 8, 2014.

[41] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[42] D. B. Rubin, "Statistical disclosure limitation," *Journal of Official Statistics*, vol. 9, no. 2, pp. 461–468, 1993.

[43] Y. Rubner, C. Tomasi, and L. Guibas, "A metric for distributions with applications to image databases," in *Computer Vision, 1998. Sixth International Conference on*, jan 1998, pp. 59 –66.

[44] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval," *International Journal of Computer Vision*, vol. 40, pp. 99–121, 2000, 10.1023/A:1026543900054. [Online]. Available: http://dx.doi.org/10.1023/A:1026543900054

[45] R. Shokri, G. Theodorakopoulos, G. Danezis, J.-P. Hubaux, and J.-Y. Le Boudec, "Quantifying location privacy: the case of sporadic location exposure," in *Proceedings of the 11th international conference on Privacy enhancing technologies*, ser. PETS'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 57–76. [Online]. Available: http://dl.acm.org/citation.cfm?id=2032162.2032166

[46] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, ser. SP '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 247–262. [Online]. Available: http://dx.doi.org/10.1109/SP.2011.18

[47] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: optimal strategy against localization attacks," in *ACM Conference on Computer and Communications Security (CCS'12)*, T. Yu, G. Danezis, and V. D. Gligor, Eds. ACM, 2012, pp. 617–627.

[48] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[49] A. Suzuki, M. Iwata, Y. Arase, T. Hara, X. Xie, and S. Nishio, "A user location anonymization method for location based services in a real environment," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '10. New York, NY, USA: ACM, 2010, pp. 398–401. [Online]. Available: http://doi.acm.org/10.1145/1869790.1869846

[50] A. Tockar, "Riding with the stars: Passenger privacy in the nyc taxicab dataset." [Online]. Available: http://research.neustar.biz/2014/09/15/riding-with-the-stars-passenger-privacy-in-the-nyc-taxicab-dataset/

[51] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *Information Theory, IEEE Transactions on*, vol. 13, no. 2, pp. 260–269, 1967.

[52] Y. Wang, D. Xu, X. He, C. Zhang, F. Li, and D. Xu, "L2p2: Location-aware location privacy protection for location-based services," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1996–2004.

[53] R. W. White, A. Hassan, A. Singla, and E. Horvitz, "From devices to people: Attribution of search activity in multi-user settings," in *Proc. International World Wide Web Conference (WWW)*, 2014.

[54] J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett, "Differentially private histogram publication," *The VLDB Journal*, vol. 22, no. 6, pp. 797–822, Dec. 2013. [Online]. Available: http://dx.doi.org/10.1007/s00778-013-0309-y

[55] M. Ye, D. Shou, W.-C. Lee, P. Yin, and K. Janowicz, "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*. ACM, 2011, pp. 520–528.

[56] T.-H. You, W.-C. Peng, and W.-C. Lee, "Protecting moving trajectories with dummies," in *Mobile Data Management, 2007 International Conference on*, May 2007, pp. 278–282.

[57] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and pois," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 186–194.

[58] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM, 2011, pp. 145–156.

[59] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 791–800. [Online]. Available: http://doi.acm.org/10.1145/1526709.1526816

APPENDIX

A. Computational Efficiency

The fake generation process, which results in a pool of fake traces having passed the privacy test, is run offline on powerful machines, before the user's device retrieves and uses such fakes. Therefore, this computational burden is not placed on the user's device. Nevertheless, the generation process is reasonably efficient. For example, with the experimental setup described in Section VI, we could generate one fake trace in less than 2 minutes, per CPU-core, using a regular laptop. Using a powerful machine, we generated thousands of fakes in a few hours. Also note that the computation of both the aggregate mobility and the semantic clustering needs to be done only once for each input set of real traces. The former's computation time is $O(SL + (RT)^2)$ where $S = |\mathcal{S}|$ is the number of seed traces, L is the length (i.e., number of events) of each seed trace. The latter is dominated by $S(S-1)$ semantic similarity computations (e.g., each taking $O(TR^3)$ in the zeroth-order case) and one clustering operation. Excluding the final clustering, this step is embarrassingly parallel: the semantic similarity for any two users u, v can be computed independently. Also, if a few input traces are added, both the aggregate statistics and the semantic clustering can be updated and do not need to be recomputed from scratch. Once the semantic clustering has been computed, an arbitrarily large number of fakes for each seed can be generated. This process is also embarrassingly parallel, since each fake can be generated independently of other fakes for that seed, and other seeds.