Protecting Spatiotemporal Event Privacy in Continuous Location-Based Services

Yang Cao ©, Yonghui Xiao, Li Xiong ©, Liquan Bai, and Masatoshi Yoshikawa ©

Abstract—Location privacy-preserving mechanisms (LPPMs) have been extensively studied for protecting users’ location privacy by releasing a perturbed location to third parties such as location-based service providers. However, when a user’s perturbed locations are released continuously, existing LPPMs may not protect the sensitive information about the user’s real-world activities, such as “visited hospital in the last week” or “regularly commuting between location A and location B every weekday” (it is easy to infer that location A and location B may be home and office), which we call it spatiotemporal event. In this paper, we first formally define spatiotemporal event as Boolean expressions between location and time predicates, and then we define \(-\) spatiotemporal event privacy by extending the notion of differential privacy. Second, to understand how much spatiotemporal event privacy that existing LPPMs can provide, we design computationally efficient algorithms to quantify the spatiotemporal event privacy leakage of state-of-the-art LPPMs. It turns out that the existing LPPMs may not adequately protect spatiotemporal event privacy. Third, we propose a framework, PriSTE, to transform an existing LPPM into one protecting spatiotemporal event privacy by calibrating the LPPM’s privacy budgets. Our experiments on real-life and synthetic data verified that the proposed method is effective and efficient.

Index Terms—Location-based services, location privacy, location obfuscation, Markov model, trajectory privacy

1 INTRODUCTION

The continued advances and usage of smartphones and GPS-enabled devices have provided tremendous opportunities for Location-Based Service (LBS), such as Yelp or Uber for snapshot or continuous queries, for example, “where is the nearest restaurant” or “continuously report the taxis within one mile of my location”. Mobile users have to share their real-time locations or a sequence of locations with the service providers, which raises privacy concerns since users’ digital trace can be used to infer sensitive information, such as home and workplace, religious places and sexual inclinations [2], [3], [4].

A large number of studies (see surveys [5], [6], [7]) have explored how to protect user’s location privacy which can be categorized from different aspects: privacy goals, adversarial models, location privacy metrics, and location privacy preserving mechanisms (LPPMs). Privacy goals indicate what should be protected or what are the secrets (e.g., a single location or a trajectory); adversarial models make assumptions about the adversaries; location privacy metrics formally define the quantitative measurement of the protection w.r.t. the privacy goal; LPPMs are designed to achieve a specified privacy metrics. For instance, Geo-Indistinguishability [8] is a location privacy metrics, which is receiving increasing attention since it extends the notion of differential privacy [9] to the location privacy setting so that the protection level does not depend on adversaries’ prior knowledge; the privacy goal of Geo-Indistinguishability is to protect a single location (can be extended for protecting location trace [10]); Laplace Planar Mechanism [8] is an LPPM satisfying Geo-Indistinguishability. Another example is Planar Isotropic Mechanism [11] for the metrics of \(\delta\)-location set privacy to protect each location in a trajectory. These state-of-the-art LPPMs take an actual location and a privacy parameter as inputs and probabilistically output a randomly perturbed location. A LPPM privacy parameter controls the location privacy level. For examples of the above mechanisms, the privacy parameter is dened as a positive real value and a smaller privacy parameter indicates stronger privacy protection. In other words, the privacy parameter can be considered as the controlled level of privacy loss.

We argue that the existing techniques may not adequately protect users’ sensitive information in their real-world activities because the privacy goal is not well-defined. Most of the existing studies only focused on the protection of either a single location or a trajectory, which does not completely reflect the secrets that should be protected in users’ real-world activities. To explain this, we need to formally define the sensitive information in the users’ real-world activities. We define a user’s a single location at time \(t\) as a predicate \(l_t = s_i\), where \(l_t\) is a variable representing the user’s position at time \(t\) and \(s_i \in S, i \in [1, m]\) is a location on the map \(S\) of \(m\) locations. The value of such predicate can be either \(true\) or \(false\), which could be a secret of the user. Then, we can generalize users’ secrets in their real-world activities as Boolean expressions of combining different predicates.
over spatial and/or temporal dimensions, which is called **spatiotemporal event** in this paper.

In Fig. 1, we illustrate six representative Boolean expressions between location and time dimensions. We use $s_1$ and $s_2$ to denote two locations on the map $S$, and use $l_1$ and $l_2$ to denote two variables about a user’s locations at timestamps $1$ and $2$, respectively. Event (a) is always false since a user cannot be physically at two different locations at the same time. Event (b) means that the secret is a sensitive region (or area) of $\{s_1, s_2\}$. Event (c) represents a sensitive trajectory $s_1 \rightarrow s_2$ between timestamps $1$ and $2$, i.e., the user stays at $s_1$ at time $1$ and $2$. Event (d) denotes that the secret is the visit to $s_1$ at timestamp 1 or 2. Event (e) depicts the secret as a type of trajectory pattern, i.e., the user may stay at two sensitive regions successively; a real-world example of such an event is "regularly commuting between Address 1 and Address 2 every morning and every afternoon", i.e., periodic spatiotemporal events may happen every week day. Event (f) indicates the secret as user’s presence in sensitive region $\{s_1, s_2\}$ at either timestamp 1 or 2; a real-world example of such an event is "visited hospital in the last week", i.e., the hospital visit may happen once or multiple times at any time in last week.

We can see that the spatiotemporal events representing sensitive locations and a trajectory (i.e., (b) and (c)), which are the major privacy goals of previous studies, are only two cases among the six enumerated examples. Hence, even if an LPPM protects each location or a trajectory, it may not protect a complex spatiotemporal event such as the ones shown in Figs. 1e and 1f since such new privacy goals have not been formalized in the literature.

In this paper, we attempt to achieve spatiotemporal event privacy by leveraging the existing LPPMs designed for conventional location privacy. There are three major challenges below. First, we lack the formal definition of spatiotemporal event and privacy metrics for it. Second, evaluating the privacy guarantee of a given spatiotemporal event could be computationally intractable since the event can be extremely complicated. Taking the pattern event (e.g., Fig. 1e) for example, if the sensitive region includes $m$ locations and the length of such event spans $T$ timestamps, there are $m^T$ possible trajectories that need to be protected, which may lead to exponential time computation. Third, similar to Geo-Indistinguishability, we hope to design a mechanism that is robust to adversaries with any prior knowledge.

**Contributions.** To the best of our knowledge, this is the first paper that studies how to achieve spatiotemporal event privacy. Our contributions are summarized as follows.

First, we study the privacy goal and privacy metrics for protecting spatiotemporal event (Section 2). We formally define a new type of privacy goal, i.e., spatiotemporal events, as Boolean expressions of location-time predicates, and propose a privacy metric, $\varepsilon$-spatiotemporal event privacy, for protecting spatiotemporal events by extending the notion of differential privacy. We also explore the difference between the metrics of location privacy and spatiotemporal event privacy. It turns out that, although the definition of spatiotemporal event is more general than a single location or a trajectory, the privacy metrics between spatiotemporal event privacy and location privacy can be orthogonal and complementary: Location privacy provides general protection against unknown risks, while spatiotemporal event privacy guarantees flexible and customizable protection which may not be provided by the existing LPPMs. Hence, it would be preferable that an LPPM achieving a location privacy metrics such as Geo-Indistinguishability can also satisfy $\varepsilon$-spatiotemporal event privacy w.r.t. user-specified events.

Second, we develop efficient algorithms for quantifying how much $\varepsilon$-spatiotemporal event privacy a given LPPM can provide w.r.t. adversaries with a specific prior knowledge about the user’s initial probability distribution over possible locations (Section 3). We model an LPPM as an emission matrix that takes user’s true position as input and outputs a perturbed location. As we mentioned previously, one of the challenges in quantifying the probability of a spatiotemporal event is that the computational complexity may be exponentially increasing with the number of predicates in a user-specified spatiotemporal event. We develop a novel two-possible-world method to quantify spatiotemporal event privacy with linear complexity.

Third, based on our quantification method, we propose a framework, i.e., PriSTE (Private Spatio-Temporal Event), which converts a mechanism for location privacy into one for spatiotemporal event privacy against adversaries with any prior knowledge (Section 4). We demonstrate the effectiveness of our framework by two case studies using state-of-the-art LPPMs, i.e., Laplace Planar Mechanism for Geo-Indistinguishability [8] and Planar Isotropic Mechanism for $\delta$-location set privacy [11].

Finally, we evaluate our algorithms on both synthetic and real-world datasets testing its feasibility, efficiency, and the impact of various parameters (Section 5).

## 2 Defining Spatiotemporal Event Privacy

### 2.1 Scenario

We consider a scenario that a single user continuously releases her perturbed location with an untrusted third party such as a location-based service provider. The user’s true locations are denoted by $l_1, l_2, \ldots, l_T$. A location privacy-preserving mechanism (LPPM) blurs user’s true location $l_t$ to a perturbed one $o_t$ that satisfies a privacy metrics such as Geo-Indistinguishability [8] or $\delta$-location set privacy.
We clarify our assumptions about LPPM and users’ mobility model as follows. First, we consider an LPPM that takes input as user’s true location \( l_t \) and outputs a perturbed location \( \bar{l} \). We use an \( m \times m \) emission matrix where each cell is the emission probability (as shown in Fig. 2a) to represent the LPPM. Second, we assume a user’s location at time \( t+1 \) is correlated with her location at time \( t \), representing by an \( m \times m \) transition matrix as shown in Fig. 2b, and such transition matrix is public information which can be learned from either historical trajectory or the pattern of road networks. We model the correlation between user’s consecutive locations using first-order time-homogeneous Markov model, i.e., the transition matrix is identical at each step.

2.2 Spatiotemporal Events

We first define location-time predicate, which is an atomic element in spatiotemporal events. Let \( S = \{s_1, ..., s_m\} \) be the domain of all possible locations, where \( m \) is the size of the domain and \( s_i \) is one location (we use state interchangeably) on the map. At time \( t \), a user’s location can be stated as \( l_t = s_i \), which means the user is at location \( s_i \) at time \( t \). We call \( l_t = s_i \) location-time predicate, whose value can be true or false depending on the ground truth of user’s state at \( t \).

We define spatiotemporal events as Boolean expressions of the location-time predicates.

Definition 2.1 (Event). A spatiotemporal event, denoted by \( E \), is a single location-time predicate or a combination of location-time predicates linked by the Boolean operators AND, OR, NOT (i.e., \( \land, \lor, \neg \)), respectively.

For the ease of exposition, we define the following notations. We denote a region (i.e., a set of locations) by a vector \( s \in \{0,1\}^{m \times 1} \) where the \( i \)th element is 1 only if the region contains \( s_i \). We use \( S \) to indicate a sequence of regions. We denote the corresponding timestamp of each region by \( T \) as a sequence of timestamps with the same cardinality of \( S \).

Using Boolean logic to define spatiotemporal events enables users to customize their privacy preference for diverse real-world activities as shown in Fig. 1. A pair of \( i \)th elements in \( S \) and \( T \) could form a single region event as shown in Event (b) in Fig. 1. These single region events could be combined by AND or OR, which form Presence or Pattern (e.g., Events (e) and (f) in Fig. 1).

2.2.1 Presence Event

When the secret is whether or not a user visited a sensitive region (e.g., medical facilities) in a given time period, we can use Presence to represent such secret. A Presence event holds if a user appears in any one of the regions with user-specified timestamps. In the simplest case of Presence, when the region includes only one location and the time period consists of one timestamp, it reduces to a single location event. Hence, Presence event can be seen as a generalization of single location event.

Definition 2.2 (Presence). Given a sequence of regions \( S = \{s_1, ..., s_k\} \) and a sequence of timestamps \( T = \{t_1, ..., t_n\} \), if a user appears in at least one \( s_k \in S \) at the corresponding time \( t_k \in T \), then it is a presence event, denoted by Presence\((S, T)\).

Example 2.1 (Example of Presence). Fig. 3 shows a map of \( S = \{s_1, s_2, s_3\} \). For this event, the region \( s = [1, 1, 0]^T \) denoting the states \( s_1 \) and \( s_2 \); the time period \( T = [3, 4] \) denoting timestamp 3 and 4. Let \( S = [s, s] \). This Presence \((S, T)\) event is expressed as \( (l_3 = s_1) \lor (l_4 = s_2) \lor (l_4 = s_1) \lor (l_2 = s_2) \). The shaded region shows a Presence event that the user appears in a region of \( \{s_1, s_2\} \) during
timestamps 3 and 4. If the user’s true trajectory passes through the shaded region (at least one timestamp), the event is true.

### 2.2.2 Pattern Event

We use Pattern to represent the secret whether or not a user visited multiple sensitive regions successively. In a simple case of Pattern event, the regions consist of locations at a sequence of timestamps, then it is reduced to single trajectory event. Hence, Pattern is a generalization of a user’s trajectories.

**Definition 2.3 (Pattern).** Given a sequence of regions $S = [s_1, ..., s_n]$ and a sequence of timestamps $T = [t_1, ..., t_n]$, if a user appears in all $\{s_1, ..., s_n\}$ sequentially at the corresponding time during $T$, then it is a pattern event, denoted by Pattern($S$, $T$).

**Example 2.2 (Example of Pattern).** The Pattern event in Fig. 3 represents trajectories with a pattern going through a sensitive region $\{s_1, s_2\}$ at timestamp 2 and the same region $\{s_1, s_2\}$ at timestamp 3 successively. This Pattern event is expressed as $\{(l_2 = s_1) \lor (l_2 = s_2)\} \land \{(l_3 = s_1) \lor (l_3 = s_2)\}$.

### 2.2.3 Discussion

From the above definitions, we can see that, in terms of privacy goal, spatiotemporal event privacy can be a generalization of location privacy studied in the literature in which the privacy goal is protecting a single location or a trajectory. In this paper, we focus on the two representative events defined above, i.e., Presence and Pattern, which are the two most complicated and unexplored events among examples in Fig. 1. We note that Presence and Pattern include the cases when the time $T$ is not consecutive. Users can specify one or multiple events to be protected.

On the other hand, it could be a non-trivial task for end-users to define a spatiotemporal event that needs to be protected. We provided a tool in our recent demonstration [12] to help users explore how accurate an adversary could infer a given event so that to identify and protect risky spatiotemporal events. Boolean logic is an expressive tool for representing spatiotemporal events which could be complicated. Besides the users burden on defining privacy preference, another negative effect (due to the expressiveness) may be that an event with complicated logic could be hard to protect with meaningful utility and reasonable runtime. We address the problem of computation complexity in Section 3.

### 2.3 c-Spatiotemporal Event Privacy

Inspired by the definition of differential privacy [9], we define c-Spatiotemporal Event Privacy as follows.

**Definition 2.4 (c-Spatiotemporal Event Privacy).** A mechanism preserves c-Spatiotemporal Event Privacy for a spatiotemporal Event if at any timestamp $t$ in $\{1, ..., T\}$ given any observations $\{o_1, ..., o_t\}$,

$$\Pr(\{o_1, ..., o_t\}|Event) \leq c^t \Pr(\{o_1, ..., o_t\}|¬Event),$$

(1)

where Event is a logic variable about the user-specified spatiotemporal event and $¬Event$ denotes the negation of Event.

There are two major benefits of extending differential privacy to protecting spatiotemporal events. First, it provides a well-defined semantics for spatiotemporal event privacy. Similar to differential privacy that requires the indistinguishability between any two neighboring databases [9], c-Spatiotemporal Event Privacy requires the indistinguishability regarding whether the Event is true or false given any observations. It provides a clear privacy semantics: it is hard for adversaries to distinguish whether the event happened or not. Another benefit is that, similar to differential privacy whose privacy guarantee is independent of the prior probability of a given database, the privacy provided by c-Spatiotemporal Event Privacy is independent of the prior probability of the protected event.

To better understand the characteristics of spatiotemporal event privacy, we illustrate the indistinguishability-based privacy metrics for the three privacy goals in Fig. 4, where the lines connecting two secrets indicate the requirements of indistinguishability between the corresponding two possible values of the secrets.

As shown in Fig. 4a, indistinguishability-based location privacy metrics (such as Geo-Indistinguishability) [8] requires indistinguishability between each pair of locations. Indistinguishability-based trajectory privacy metrics [10], [11], [12], [13] requires indistinguishability between each pair of possible trajectories as shown in Fig. 4b. Whereas, c-Spatiotemporal event privacy requires indistinguishability between the defined event and its negation. For example, if the spatiotemporal event is defined as Pattern($S$, $T$) where $S = [s_1, s_2]$, $s_1 = \{s_1\}$, $s_2 = \{s_1, s_2, s_3\}$ and $T = [1, 2]$ (i.e., a trajectory passes through $s_1$ and then a region $\{s_1, s_2, s_3\}$ successively), then it only requires the indistinguishability between the set of all possible trajectories that pass through $\{s_1\}$ and $\{s_1, s_2, s_3\}$ and the set of trajectories that do not. This spatiotemporal event privacy makes sense when some mobility patterns are sensitive. For example, if $s_1$ is “hospital”, $s_2$ is “home”, and $s_3$ is “office”, the pattern from $s_1$ to $\{s_1, s_2, s_3\}$ could be sensitive.

We note that spatiotemporal event privacy is orthogonal to location privacy or trajectory privacy. First, protecting the privacy of a single location or a trajectory may not imply the protection of spatiotemporal event privacy because spatiotemporal event can be complex as shown in Figs. 1e or 1f. The existing LPPMs are designed to ensure privacy metrics defined on locations or trajectories. One of our focus in this study is to quantify how much spatiotemporal event privacy a given LPPM can provide, which will be elaborated in the next section. Second, protecting spatiotemporal event
privacy does not imply the protection of location privacy because they define indistinguishability over different levels of secrets. Taking Fig. 4c for example, the indistinguishability between s1 → s1 and s1 → s2 is not required in such spatiotemporal event privacy guarantee; however, it is required in trajectory privacy as shown in Fig. 4b. Even if we define the event in spatiotemporal event privacy as a single location, say s1, the guarantee of spatiotemporal event privacy is the indistinguishability between s1 and \{s2, s3\}, which does not guarantee the indistinguishability between s1 and s2.

It would be preferable if we achieve both location privacy and spatiotemporal event privacy so that a user can enjoy the best of two worlds: Location privacy provides general protection against unknown risks when sharing location with the third parties, while spatiotemporal event privacy guarantees customizable protection which may prevent against profiling attacks [3], [14]. Therefore, in this paper, we study how to use an existing probabilistic LPPM (e.g., Laplace Planar Mechanism [8] and Planar Isotropic Mechanism [11]) to achieve ε-spatiotemporal event privacy.

We note that the definition of events may reveal a user's sensitive information. In this paper, we assume that the events and the protection mechanisms are locally and securely stored in the user's device. The user may specify one or multiple events that need to be protected. In practice, we can also have default that are suggested by a privacy preference recommendation system for users' selection [15] or pre-specified event templates that are given by the user.

### 3 Quantifying Spatiotemporal Event Privacy

#### 3.1 Overview of Our Approach

For ease of exposition, we first assume that adversaries who have a specific knowledge of the user's initial probability distribution over possible locations, which is denoted by \(\pi\), as in the next section, we will remove this assumption so that the spatiotemporal event privacy leakage will be bounded in ε w.r.t. adversaries with any prior knowledge of user's initial probability.

Now, we explain the main goal of quantifying the spatiotemporal event privacy leakage of the LPPM and our approach. Based on Definition 2.4 of ε-spatiotemporal event privacy, we need to calculate the maximum ratio of \(\Pr(s_0, s_1, \ldots, s_T | \text{Event})\) in which \(s_0, s_1, \ldots, s_T\) are released by a given LPPM. This ratio can be considered as spatiotemporal event privacy leakage w.r.t. the user-specified event. We quantify this ratio w.r.t. given observations \(s_0, s_1, \ldots, s_T\) and a given user's initial probability \(\pi\), so that we can directly calculate the \(\Pr(s_0, s_1, \ldots, s_T | \text{Event})\). In Section 4, we will design a mechanism for spatiotemporal event privacy w.r.t. any observations and arbitrary initial probability. Our goal in this section is to calculate the likelihood of the observations given Event or \(\neg \text{Event}\), i.e., \(\Pr(s_0, s_1, \ldots, s_T | \text{Event})\) or \(\Pr(s_0, s_1, \ldots, s_T | \neg \text{Event})\), which can be derived by \(\Pr(s_0, s_1, \ldots, s_T | \text{Event}) = \frac{\Pr(s_0, s_1, \ldots, s_T, \text{Event})}{\Pr(\text{Event})}\). We call \(\Pr(\text{Event})\) as prior probability of the event, and \(\Pr(s_0, s_1, \ldots, s_T, \text{Event})\) as joint probability of the event.

A severe challenge of calculating the prior or joint probabilities of the event is the computational complexity. Given an arbitrary spatiotemporal event, we need to enumerate all possible combination of the Boolean expression for prior and joint probabilities, which can be exponential to the number of predicates in the expression. To address this problem, we propose a two-possible-world method for computing the prior and joint probabilities in Sections 3.2 and 3.3.

#### For ease of exposition, we define notations below. \(M \in \mathbb{R}^{m \times m}\) denotes a transition matrix that describes temporal correlations in user's location. At timestamp 1, an initial probability is denoted by \(\pi \in \{0, 1\}^{1 \times m}\). During timestamp \(\{1, 2, \ldots, T\}\), the probability of the true location \(\Pr(l_t)\) is denoted by a row vector \(p_t \in \{0, 1\}^{1 \times m}\) where \(t\)th element denotes \(\Pr(l_t = s_t)\). A Markov model follows the transition property of \(p_{t+1} = p_M \cdot p_t\), e.g., after a Markov transition, \(p_2 = \pi M\) at timestamp 2 given \(p_1 = \pi\).

The notations below for matrix computation are also used in the rest of this paper. Let 0 and 1 be row vectors with \(m\) elements being 0 and 1 respectively. \([0, 1]\) is a row vector in \(\mathbb{R}^{1 \times m}\), \(a \circ b\) denotes the Hadamard product of \(a\) and \(b\). \(a^D\) is a diagonal matrix with the elements of vector \(a\) on the diagonal.

#### 3.2 Computing Prior Probability of an Event

To avoid the exponential complexity, we propose an efficient algorithm with two possible worlds denoting whether the Event is true or false. The states in the two worlds denote the joint probabilities \(\Pr(l_t = s_t, \text{Event})\) and \(\Pr(l_t = s_t, \neg \text{Event})\). For Presence, once a trajectory enters into the region of the Presence, its probability will be kept in the world of true Event forever. For Pattern, the probability distribution among the two worlds are derived at the beginning timestamp of the Event, and only the trajectories satisfying the Pattern will be kept in the world of true Event. At last, the sum of probabilities in the world of true Event will be \(\Pr(\text{Event} = \text{true})\).

In the following, we study how to compute the prior probabilities of Presence and Pattern events. For simplicity, we view the events in the rest of the paper are defined in consecutive time and use \(\text{start}\) and \(\text{end}\) to denote the start time point and end time point of the user-specified spatiotemporal event. We assume \(S\) to be \(\{s_1, s_2, s_3\}\) in the following examples.

#### 3.2.1 Presence Events

**Example 3.1.** Let us consider the same Presence event defined in Example 2.1, It is defined as an event passing through \(s_1\) or \(s_2\) during \(t = 3\) or \(t = 4\), i.e., \(s = \{1, 1, 0\}^T\), \(\text{start} = 3, \text{end} = 4\). The transition matrix \(M\) is given below:

\[
M = \begin{bmatrix}
0.1 & 0.2 & 0.7 \\
0.4 & 0.1 & 0.5 \\
0 & 0.1 & 0.9
\end{bmatrix}
\]
Fig. 5. New Markov transitions: all transitions going to the Presence region will be re-directed to the virtual worlds.

Then Fig. 5 shows the new transitions in the two worlds, the top world and the bottom world separated by the dashed line in Fig. 5, corresponding to the two possible worlds where the presence event is false or true respectively. From time 1 to 2, a normal transition can be made. At timestamp 2, all the transitions going to the states \(s_1\) and \(s_2\) will be re-directed to the new states \(s'_1\) and \(s'_2\), denoting the states when the Presence happens. Other transitions that do not go to the region will be performed normally. Similarly at time 3, the transition from \(s_3\) to \(s_2\) will also go to the state \(s'_3\) because the event is also true in this case. After time 4, the original Markov transitions come back to work again.

The intuition can be formalized as follows. First, the original probability \(\pi\) becomes \([\pi, 0]\). Second, the transition matrix \(M_t\) takes the form of four transition matrices between the two virtual worlds, i.e., the Event is true or false, in Eq. (3).

Then the new transition matrix can be derived in Eqs. (4) and (5) for different time period where \(M\) is the original transition matrix and \(s^D\) is the diagonal of the region \(s\) of Presence defined in Definition 2.2.

![Fig. 5.](image)

**Example 3.2.** We study the Pattern event as illustrated in Fig. 6. At time 1, the transitions entering \(s_1\) and \(s_2\) go to \(s'_1\) and \(s'_2\). From time 2 to 4, the transitions in the top world were performed normally. However, the transitions from the bottom world go back to the top world if the destinations are not in the defined regions. At time 5, the original Markov transitions come back to work again.

From above example, the transition matrices for Pattern differ from the ones for Presence during the event time from \(\text{start to end} - 1\) (i.e., Eq. (7)). On the other hand, when it is outside the event, i.e., \(t < \text{start} - 1\) or \(t \geq \text{end}\), the transition matrices for Pattern are the same as the ones for Presence (i.e., the matrices in (8) and (5) are the same). Finally, when \(t = \text{start} - 1\), the transition matrices for Pattern is also as same as the ones for Presence (i.e., the matrices in (6) and (4) are identical)

\[
M_t = \begin{bmatrix} M - Ms^D & Ms^D \\ 0^D & M \end{bmatrix}, t = \text{start} - 1. 
\]

(6)

\[
M_t = \begin{bmatrix} M - Ms^D & Ms^D \\ 0^D & M \end{bmatrix}, \text{start} \leq t \leq \text{end} - 1. 
\]

(7)

\[
M_t = \begin{bmatrix} M & 0^D \\ 0^D & M \end{bmatrix}, t < \text{start} - 1 \text{ or } t \geq \text{end}. 
\]

(8)

In summary, the prior probability of any Event can be derived as the sum of probabilities in the world where the Event is true. Lemma 3.1 shows the formal computation.

**Lemma 3.1.** For initial probability \(\pi \in \mathbb{R}^{1 \times m}\), the prior probability of an Event of Presence and Pattern is

\[
\Pr(\text{Event}) = [\pi, 0] \prod_{i=1}^{\text{end} - 1} M_i [0, 1]^T, 
\]

(9)

where \(M_i\) is computed by Eqs. (4), (5), (6), (7), (8).

If the Markov model is time-varying, i.e., when the transition matrices \(M\) at different \(t\) are not identical, the only extra effort is to re-compute Eqs. (4)–(8) using the corresponding transition matrix \(M\) at \(t\).

### 3.3 Computing Joint Probability of an Event

The calculation of a spatiotemporal event and a sequence of observed locations, i.e, \(\Pr(o_1, o_2, \ldots, o_T, \text{EVENT})\) is a little
more complicated than previous sections since it depends on not only the initial probabilities but also the emission matrix of the LPPM. Similarly, we use two-possible-world method to avoid enumerating all possible cases of an event. We utilize forward-backward algorithm[16] to estimate the probability of the true state (true location) at timestamp \( t \) given all observations \( \mathbb{P}(l_t|o_1, o_2, \ldots, o_T) \). It first calculates a forward probability \( a^k_t = \mathbb{P}(o_t|l_t = s_k) \) iteratively, i.e.,

\[
\alpha^k_t = \mathbb{P}(o_t|l_t = s_k) = \sum_{i=1}^{M} \alpha^k_{i-1} \mathbb{P}(l_t = s_k|l_{t-1} = s_i), \tag{10}
\]

Then, a backward probability \( \beta^k_t = \mathbb{P}(o_{T-t} \cap \ldots \cap o_2|l_t = s_k) \) can also be derived by

\[
\beta^k_t = \sum_{i=1}^{M} \mathbb{P}(l_{t+1} = s_i|l_t = s_k) \beta^k_{i+1} \tag{11}
\]

By initializing \( \beta^k_1 = 1 \) for all \( k \), we can obtain the estimation of \( l_t \) as follows:

\[
\mathbb{P}(l_t = s_k|o_1, o_2, \ldots, o_T) = \frac{\alpha^k_t \beta^k_t}{\sum_{i=1}^{M} \alpha^k_i \beta^k_i}. \tag{12}
\]

**Intuition.** The intuition of our solution is to use the forward-backward algorithm in the two virtual worlds where the event is true and false. This is feasible because the emission probability, which determines the probabilities of the observations, is independent from any events. Hence in our computation the forward probability and backward probability are \( \mathbb{P}(\text{Event}, o_1, o_2, \ldots, o_T) \) for \( t \leq \text{end} \) and \( \mathbb{P}(\text{Event} \cap o_1, o_2, \ldots, o_T|\text{Event}) \) for \( t > \text{end} \) respectively. By combining them together, we can obtain the posterior probability of the event. Note that at any timestamp \( t \leq \text{end} \), we do not see the future (\( t > \text{end} \) observations). Thus the posterior probability only counts to the current timestamp \( t \).

**Before and During the Event.** In the forward algorithm, the probability \( \alpha^k_t = \mathbb{P}(l_t = s_k, a_1, o_2, \ldots, a_t) \) is derived at timestamp \( t \). We represent \( \alpha^k_t \) in the vector form \( \mathbf{a}_t = [\alpha^k_1, \alpha^k_2, \ldots, \alpha^k_n] \). Then it can be derived as \( \mathbf{a}_t = (\mathbf{a}_{t-1} \odot \mathbf{M}_{t-1}) \odot \mathbf{p}_t \). Without any further observations, the joint probability can be derived from Lemma 3.1. The result is shown in Lemma 3.2.

Lemma 3.2. Given an initial probability \( \pi \), the joint probability of an Event of Presence or Pattern and observations \( o_1, o_2, \ldots, o_T \) at any timestamp \( t \leq \text{end} \) is

\[
\mathbb{P}(\text{Event}, o_1, o_2, \ldots, o_T) = \|\pi \|_0 \left( \prod_{t=2}^{T} \mathbf{M}_{t-1} \mathbf{p}_t \prod_{t=1}^{T} \mathbf{M}_{t} \|0, 1\|_T \right)^T. \tag{13}
\]

**After the Event.** In the backward algorithm, \( \beta^k_T = \mathbb{P}(a_T \cap \ldots \cap a_2, o_T|l_t = s_k) \). We represent it in the vector form \( \beta_t = [\beta_1^k, \beta_2^k, \ldots, \beta_T^k] \). Then it can be derived as \( \beta_t = (\beta_{t+1} \odot \mathbf{p}_{t+1} \mathbf{M}_{t} \odot \mathbf{p}_{t+1} \mathbf{M}_{t}) \). Similarly, we have Lemma 3.3 for joint probability.

Lemma 3.3. Given an initial probability \( \pi \), the joint probability of an Event of Presence or Pattern and observations \( o_1, o_2, \ldots, o_T \) at any timestamp \( t > \text{end} \) is

\[
\mathbb{P}(\text{Event}, o_1, o_2, \ldots, o_T|\text{Event}) = \|\pi \|_0 \left( \prod_{t=2}^{T} (\mathbf{M}_{t-1} \mathbf{p}_t) \prod_{t=1}^{T} \mathbf{M}_{t} \|0, 1\|_T \right)^T. \tag{14}
\]

To summarize, now we can quantify the ratio \( \mathbb{P}(o_1, o_2, \ldots, o_T|\text{Event}) = \frac{\mathbb{P}(o_1, o_2, \ldots, o_T|\text{Event})}{\mathbb{P}(\text{Event})} \) for spatiotemporal event privacy using Lemma 3.1 to compute \( \mathbb{P}(\text{Event}) \) and Lemmas 3.2, 3.3 to compute \( \mathbb{P}(o_1, o_2, \ldots, o_T|\text{Event}) \). We note that our approach of computing the joint probability of an event is able to deal with different emission matrices at each \( t \). Since \( \mathbf{p}_t \) is a vector of emission probabilities given the observation \( o_t \), i.e., a column in the emission matrix, and \( \mathbf{p}_D \) is a diagonal matrix whose diagonal elements are \( \mathbf{p}_t \), we only need to obtain \( \mathbf{p}_t \) and \( \mathbf{p}_D \) from the corresponding emission matrix at \( t \) and then use such \( \mathbf{p}_D \) in Eqs. (13) and (14).

## 4 PriSTE Framework

In previous section, we designed methods for quantifying \( \epsilon \)-spatiotemporal event privacy provided by an LPPM w.r.t. a specified initial probability, which means that the privacy loss may not be bounded within \( \epsilon \) if an attacker has a different initial probability.

In this section, we first design the Private Spatio-Temporal Event (PriSTE) framework and then solve the above problem by checking if \( \epsilon \)-spatiotemporal event privacy for any initial probabilities. Finally, we demonstrate two case studies that instantiate the framework based different location privacy metrics for protecting spatiotemporal event privacy.

### 4.1 PriSTE

Based on the quantification techniques that we developed in previous sections, we propose a framework that converts a location privacy protection mechanism into one protecting spatiotemporal event privacy. The PriSTE framework is illustrated in Fig. 7 and described in Algorithm 1.

The major components are **Quantification** and a given LPPM. Their interactions are described as follows. First, the LPPM generates a perturbed location from the true location (Line 2 in Algorithm 1) and pass it to the **Quantification** component. By Theorem 4.1, the **Quantification** component (Line 3) checks whether this perturbed location satisfies the ratio in Eq. (1) (i.e., \( \epsilon \)-spatiotemporal event privacy), under a sequence of previous observations and user-specified spatiotemporal events. If not, we need to calibrate the emission matrix to ensure that it satisfies \( \epsilon \)-spatiotemporal event privacy. The strategy of emission matrix calibration is...
Algorithm 1. PriSTE Framework

Require: true location, \( \epsilon \), LPPM, M, Events
1: for \( t \in \{1, 2, \ldots, T\} \) do
2: generate \( o_t \) with LPPM w.r.t. the true location;
3: while \( \epsilon \)-Spatiotemporal Event Privacy not hold do
4: calibrate LPPM and generate \( o_t \);
5: end while
6: release \( o_t \);
7: end for

4.2 Privacy Checking With Arbitrary Initial Probability

According to the quantification in Section 3, we can calculate \( \Pr(o_{t_1}, o_{t_2}, \ldots, o_{t_T} | \text{Event}) \) given \( o_{t_1}, o_{t_2}, \ldots, o_T \) and a given initial probability \( \pi \). In this section, we show how to make sure the ratio is bounded given arbitrary initial probability.

Our idea is taking \( \pi \) as a variable and solving the maximization problem of \( \Pr(o_{t_1}, o_{t_2}, \ldots, o_{t_T} | \text{Event}) \) - \( \epsilon^e \). We want to make sure the maximum value is always less than or equal to 0, i.e., the user enjoys plausible deniability for her specified spatiotemporal event.

The following theorem shows the conditions related to \( \pi \) that satisfies \( \epsilon \)-spatiotemporal event privacy. We will formulate it as an optimization problem.

Theorem 4.1. For an Event of Presence or Pattern and an arbitrary initial probability \( \pi \), \( \epsilon \)-spatiotemporal event privacy is satisfied at any timestamp \( t \), i.e., \( \Pr(o_{t_1}, o_{t_2}, \ldots, o_{t_T} | \text{Event}) \leq \epsilon^e \), if the observation \( o_t \) is released based on the following two conditions

\[
\pi(1^D, 0^D)(|\epsilon - 1|) + \epsilon^e \leq \epsilon^e \]

(15)

\[
\pi(1^D, 0^D)(|\epsilon - 1|) + \epsilon^e \leq \epsilon^e \]

(16)

where

\[
a = \prod_{i=1}^{t-2} M_i[0, 1]^T.\]

For \( t \leq \text{end} \)

\[
b^T = \prod_{i=1}^{t-2} (M_i - 1) \prod_{i=1}^{t-2} [0, 1] = \prod_{i=1}^{t-2} (M_i - 1) \prod_{i=1}^{t-2} [0, 1].\]

(18)

For \( t > \text{end} \)

\[
b^T = \prod_{i=1}^{t-2} (M_i - 1) \prod_{i=1}^{t-2} [0, 1] = \prod_{i=1}^{t-2} (M_i - 1) \prod_{i=1}^{t-2} [0, 1].\]

(19)

Quadratic Programming. To determine whether Eqs. (15) and (16) are true or not for arbitrary \( \pi \), we transform them to maximization problems: finding the maximum values of the left parts of Eqs. (15) and (16) under the constraints of \( 0 \leq p_i \leq 1 \) where \( p_i \in \pi \). As long as one maximum value is larger than 0, we know that the LPPM (emission matrix) may not satisfy \( \epsilon \)-spatiotemporal event privacy. The maximization are equivalent to quadratic programing problem since they can be rewritten in a form of \( \sum \pi(A^T + A^T) / \pi^T \) where \( A \) is a matrix. We skip the computation details about solving such quadratic programing problem since many methods and tools have been proposed in literature.

In the experiments, we use IBM CPLEX optimizer [17] as our computation engine.

4.3 Case Study 1: PriSTE With Geo-Indistinguishability

In this section, we instantiate PriSTE framework using \( \alpha \)-Planar Laplace mechanism (\( \alpha \)-PLM) which is designed for Geo-Indistinguishability[8]. We first show the computation details for quantifying \( \epsilon \)-spatiotemporal event privacy by Theorem 4.1, and then design a greedy strategy for approximately achieving \( \epsilon \)-spatiotemporal event privacy.

Algorithm Design. To implement the quantification component, we need to (1) compute the internal parameters \( a, b \) and \( c \) shown in Theorem 4.1, and (2) design a strategy to calibrate the emission matrix.

For the calibration strategy for Planar Laplace Mechanism (PLM) with a specified privacy budget \( \alpha \) (which solely determines the shape of the output distribution), we exponentially decay its privacy budget because a smaller privacy budget implies stronger protection for location privacy and less information disclosure. In our algorithm, decay rate \( \alpha \) for the privacy budget in Line 19 of Algorithm 2 is a tunable parameter that provides a trade-off between efficiency and utility of the released locations. Setting a small value allows the algorithm converge faster, but at the cost of over-perturbing the location at each timestamp. In contrast, using a large value is less efficient but allows better utility.

A natural question is whether we can always find an \( \alpha \) to release a perturbed location that satisfies Eq. (1). The answer is affirmative because \( \alpha \) converges exponentially to 0. When \( \alpha = 0 \), it releases no useful information about the true location, i.e., uniformly returning a random location without using user’s true position. It is easy to verify that the Eqs. (15) and (16) are always true in this situation.

Algorithm 2 shows the computation process. To boost the efficiency of our algorithm, we use intermediate matrices \( A \) and \( B \) to facilitate the computation of \( b \) and \( c \). At time 1, we initialize the variables as line 4 ~ 8. At any time before and inside the Event, we compute the variables as line 10 ~ 11. At any timestamps after the Event, the variables are derived as line 13 ~ 14. Then we use quadratic programming methods to check Eqs. (15) and (16) to decide whether to release the \( o_t \) or not. If not, we generate a new \( o_t \).
Algorithm 2. PriSTE With Geo-Indistinguishability

Require: $\epsilon$, Event, $\alpha$-PLM, $M_i$, $\forall i \in \{1, 2, \ldots, T\}$
1: for $t$ in $[1, 2, \ldots, T]$
2: $\alpha_t \leftarrow \alpha$-PLM; >> initial budget = $\alpha$
3: if $t = 1$ then
4: $a^t \leftarrow \prod_{i=1}^{t-1} M_i [0, 1]^T$
5: $A \leftarrow I$ >> identity matrix
6: $B \leftarrow I$
7: $b^t \leftarrow \pi^0 a^t$
8: $c^t \leftarrow \pi^0 b^t$
9: else if $t < end$ then
10: $b^t \leftarrow AM_{t-1} \pi^D \prod_{i=t}^{end} M_i [0, 1]^T$
11: $c^t \leftarrow AM_{t-1} \pi^D [1, 1]^T$
12: else
13: $b^t \leftarrow A \left(\prod_{i=1}^{t} P(D) M_i [0, 1]^T \right)$
14: $c^t \leftarrow A \left(\prod_{i=1}^{t} P(D) M_i [0, 1]^T \right)$
15: end if
16: if Eqs. (15) and (16) hold then $\epsilon$ is used here.
17: release $\alpha_t$; >> okay to release $\alpha_t$
18: else
19: $\alpha \leftarrow \frac{\alpha}{2}$ go to Line 2; >> halve the budget
20: end if
21: if $t < end$ then $A \leftarrow AM_{t-1} \pi^D$
22: update $A$ by the real $\alpha_t$
23: else
24: $B \leftarrow \pi^D M_i^{-1} B$
25: update $B$ by the real $\alpha_t$
26: end if
27: end for

Complexity. The internal parameters $a$, $b$ and $c$ in Algorithm 2 need $O(mT)$ time to be evaluated. The major computational cost lies in the quadratic program for checking Eqs. (15) and (16). The complexity will be determined by the quadratic matrix $[1^D, 0^D]^T a^T c [1^D, 0^D]^T$. If it is positive definite, then the complexity is $O(m^3)$. Otherwise, with any negative eigenvalues, it will be NP-hard [18]. In our experiments, we use IBM CPLEX which can provide globally optimal results for quadratic program but may need a long computation time. We use a conservative release strategy to remedy this: we use a threshold to limit the computation time of quadratic program for checking Eqs. (15) and (16). It will not release a perturbed location unless the equations are true. Although it may lead to suboptimal solution in budget calibration, it always guarantees $\epsilon$-spatiotemporal event privacy since every released locations satisfy Eqs. (15) and (16).

Privacy Analysis. PriSTE framework relies on a local model, i.e., the assumption that adversaries cannot obtain user’s locally stored information as shown in Fig. 7. Although line 2 may be executed more than once at a timestamp $t$, Algorithm 2 still satisfies $\alpha$-Geo-Indistinguishability where $\alpha$ is the final privacy budget used for releasing $\alpha_t$ because that is the only observation of attacker at time $t$. If we remove the assumption of local model, the above statements may not be true since attacker may observe the internal states of the algorithm (which is the privacy goal of pan-privacy [19]). Examples of internal states include multiple $a_t$ tested at $t$ or the final $\alpha$ used in the algorithm. Another assumption that may affect the privacy guarantee is the transition matrix $M$, which we use it to model the correlations between locations and assume that it is given. It is an interesting future work to quantify the change of privacy loss in terms of $\epsilon$-spatiotemporal event privacy if the ground truth of correlation is not the modeled one.

4.4 Case Study 2: PriSTE With $\delta$-Location Set Privacy

To evaluate the effectiveness of PriSTE under different location privacy protection mechanisms, we also instantiate it using another privacy metrics $\delta$-location set privacy[11, 20], which is proposed for obtaining better utility by taking advantage of temporal correlation between consecutive locations in user’s trajectory. The key idea is that hiding the true location in any impossible locations (e.g., whose probabilities are close to 0) is a lost cause because the adversary already knows the user cannot be there. In other words, it restricts the output domain of the emission matrix to $\delta$-location set, which is a set containing minimum number of locations that have prior probability sum no less than $1 - \delta$. A larger $\delta$ indicates weaker privacy guarantee.

The privacy metrics of $\alpha$-Geo-Indistinguishability and $\delta$-location set privacy are orthogonal because the former requires a specific “shape” of emission distribution and the latter restricts output domain of the emission distribution. In [11], Xiao and Xiong proposed a framework to achieve $\delta$-location set privacy using a given LLPVM. For ease of comparison, we use $\alpha$-PLM as the underlying LLPVM for $\delta$-location set privacy.

Algorithm 3. PriSTE With $\delta$-Location Set Privacy

Require: $\epsilon$, Event, $\alpha$-PLM, $M_i$, $\forall i \in \{1, 2, \ldots, T\}$, $\pi$, $\delta$, $M_i$.
1: for $t$ in $[1, 2, \ldots, T]$
2: $p^t \leftarrow M_i [\pi]$; >> Markov transition
3: Construct $\Delta X_t$; >> $\delta$-location set
4: $\alpha_t \leftarrow \alpha$-PLM within $\Delta X_t$; $\delta$-location set
5: the same as Lines 3 $\sim$ 15 in Algorithm 2;
6: if Eqs. (15) and (16) hold then $\epsilon$ is used here.
7: release $\alpha_t$; >> okay to release $\alpha_t$
8: Derive posterior probability $p^t_\pi$ by Eq. (21);
9: else
10: $\alpha \leftarrow \frac{\alpha}{2}$ go to Line 4; >> halve the budget
11: end if
12: the same as Lines 21 $\sim$ 25 in Algorithm 2;
13: end for

In Line 2, when $t = 1$, we have $p^t_\pi = \pi$. In Line 8, according to [11], the posterior probability can be calculated by Eq. (21) where $p^t_\pi[i]$ and $p^t_\pi[i]$ are $i$th elements in the corresponding probability vectors

$$p^t_\pi[i] = \Pr(l_t = s_i | o_t) = \frac{\Pr(o_t | l_t = s_i) \times p^t_\pi'[i]}{\sum_j \Pr(o_t | l_t = s_j) \times p^t_\pi'[j]}.$$

Hence, we need the initial probability $\pi$ in order to calculate $\delta$-location set. In experiments, we set $\pi$ to a uniform
distribution for the evaluation of δ-location set privacy. We note that PriSTE is agnostic to such initial probability since it guarantees spatiotemporal event privacy against adversaries with arbitrary knowledge about the initial probability.

5 EXPERIMENTAL EVALUATION
In experiments, we verified that Algorithms 2 and 3 can adaptively calibrate the privacy budget of Planar Laplace Mechanism (PLM) at each timestamp for both location privacy and spatiotemporal event privacy. Especially, we highlight the following empirical findings.

- A stricter LPPM can satisfy a certain level of spatiotemporal event privacy without any change (i.e., no need of privacy budget calibration), whereas a more loose LPMM may need to reduce its privacy budget significantly for protecting the same event.

- For achieving the same level of ϵ-spatiotemporal event privacy using different LPPMs, a stricter LPMM is not always better in terms of data utility.

- If the user’s transition matrix has a significant pattern, an LPMM may need a small privacy budget to achieve ϵ-spatiotemporal event privacy.

5.1 Experiment Settings and Metrics

Dataset. We used real-life and synthetic datasets in experiments. Geolife data [21] was collected from 182 users in a period of over three years. It recorded a wide range of users’ outdoor movements, represented by a series of tuples containing latitude, longitude and timestamp. The user’s entire trajectory is used to train the transition matrix M, e.g., with R package “markovchain”.

We generated a synthetic trajectory and its transition probability matrix as follows. First, a map with 20 × 20 cells is generated. Then, the transition probability from one cell to another is drawn from the two-dimensional Gaussian distribution with scale parameter σ based on the distance between the cells. Here, a smaller σ indicates that the user moves to the adjacent cells with a higher probability, i.e., the transition matrix has a more significant pattern. Finally, we produced trajectories with 50 timestamps using such transition matrix to simulate movement of a user.

Quadratic Programming. We use the IBM CPLEX optimizer 12.7.1 [17] to find the globally optimal solution for the quadratic programming in Algorithm 2. We adopt a strategy of conservative release as mentioned previously and limit the computation time for each optimization to 1 second.

Events. We investigate Presence and Pattern events, which are represented by two parameters S and T. For example,
reducing budgets significantly. The standard deviation is higher because the perturbation of the LPPM is enhanced with a smaller \( \alpha \).

Comparing Fig. 8 with Fig. 9, where the events are defined on time periods 4–8 and 16–20 respectively, we can see that privacy budgets tend to be reduced during the defined time periods. This indicates that the final \( \alpha \) used by PLM at each timestamp may disclose the definition of spatiotemporal event as we discussed in Section 4.3. Hence, a local model is needed for PriSTE framework.

**Protecting Multiple Events.** Fig. 10 depicts the utilities when protecting two events sequentially using Algorithm 2. We can see that the utility is much worse than protecting each single event in Figs. 8 or 9 because the algorithm needs to simultaneously check if \( \epsilon \)-spatiotemporal event privacy is satisfied for both events at each time. If no perturbed location satisfying the privacy requirement of both events simultaneously, the algorithm needs to halve the privacy budget until finding an appropriate output.

**PriSTE with \( \delta \)-Location Set Privacy.** In Fig. 11, we show the utility of PriSTE with LPPMs that satisfies \( \delta \)-Location Set Privacy (Algorithm 3). Comparing Fig. 11 with Fig. 8, although both of them are using 0.2-PLM, the essential difference between them is the privacy metric: the former satisfies \( \delta \)-location set privacy and the latter satisfies Geo-Indistinguishability, i.e., 0.2-PLM in Fig. 11 has a constrained output domain. We can see that such 0.2-PLM in Fig. 11 has to reduce more privacy budgets to achieve \( \epsilon \)-spatiotemporal event privacy. Intuitively, it is because the privacy metrics of \( \delta \)-location set privacy implies a weaker privacy guarantee and its LPPM has to be stricter (using a smaller privacy budget) for protecting the event.

### 5.3 Utility Over Timestamps

In this section, we demonstrate the utility against different factors on the Geolife data and synthetic data. Figs. 12 and 13 are for protecting Presence event. Due to the space limitation, the results of protecting Pattern event are included in Appendices, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2019.2963312. Different from the utility in previous section which is averaged at each time, this section displays the utility that is further averaged over timestamps.

Hence, in the left parts of Figs. 12 and 13 (ave. budget), the steeper lines indicate the budget may be reduced heavily at some timestamps. Generally, the utility increases with a larger \( \epsilon \) in Figs. 12 and 13.

**Utility versus \( \alpha \)-Geo-Indistinguishability.** In Fig. 12, we can see that a larger \( \alpha \)-PLM needs to be calibrated heavily (i.e., has a steeper budget) for a small \( \epsilon \). Interestingly, PLMs with larger average budgets (in the left figures) may not necessarily have a better utility in terms of euclidean distance. For example, at \( \epsilon = 0.5 \), the euclidean distance of 5-PLM and 3-PLM are very close; at \( \epsilon = 1 \) or 2, 0.5-PLM and 1-PLM appear to have almost the same euclidean distance. The reason is that PLMs that have larger average budgets may have extremely small budgets at some timestamps, which results in the higher average euclidean distance over timestamps.

**Utility versus \( \delta \)-Location Set Privacy.** In Fig. 13, we can see that a PLM with a larger \( \delta \) tends to have a smaller average budget. It is because the PLM with a larger \( \delta \) indicates a weaker privacy metrics. Hence, the PLM needs to be stricter (i.e., a small budget) to achieve spatiotemporal event privacy. However, such PLM may have a better utility in terms of euclidean distance as shown in right figure of Fig. 13. The reason is that \( \delta \)-location set privacy with a larger \( \delta \) restricts the output domain significantly, which makes perturbed location close to the true location with a high probability.

The results are in line with the main purpose of \( \delta \)-location set privacy: to achieve a better privacy-utility trade-off.

**Utility versus Transition Matrices.** We compare the utility against transition matrices that have different strength of mobility patterns. As we explained previously, a smaller \( \sigma \) indicates a more significant mobility pattern. Fig. 14 shows that, for the same LPPM, it is hard to protect a spatiotemporal
event if user’s mobility pattern is significant, i.e., the LPPM needs to be very strict by using a small privacy budget. We also observe that there is no best LPPM for all ε-spatiotemporal event privacy in terms of euclidean distance.

5.4 Runtime
We name the size of $T$ and the size of $S$ as event length and event width, respectively. We also report the performance evaluation on conservative release described in Section 4.3.

Runtime versus Event Length. We fix the event width as 5 and test 100 random events with length ranging from 5 to 15. Fig. 15 shows that the average runtime of the baseline is exponential to event length and the runtime of our method is linear to the event length.

Runtime versus Event Width. We fix the event length as 5 and test 100 random events with width ranging from 5 to 15. Fig. 15 shows that the average runtime of the baseline is exponential to the event width, while our method is polynomial to the event width, which is in line with the complexity of $O(n^3)$.

Runtime versus Conservative Release. In Line 16 of Algorithm 2, we set a threshold runtime in solving the quadratic program. We do not release the perturbed location unless we are sure that Eqs. (15) and (16) are true. The threshold is a trade-off between runtime and utility as shown in Table 2 among 100 runs. We note that each runtime in Table 2 includes the whole process of Algorithm 2. In our implementation, we set the threshold to 1 second. We can see as the threshold increases, the number of conservative releases decreases, which results in increasing runtime. On the other hand, the calibrated privacy budget increases as the threshold increases. This verifies the trade-off between runtime and utility that can be achieved by the strategy of conservative release.

6 RELATED WORKS
6.1 Location Privacy Preserving Mechanisms
Existing works on location privacy can be roughly classified into two categories. The first type is the aggregated setting that guarantees differential privacy. This is different from our problem setting of an individual user in location-based applications. The second type is the individual setting, which is also our setting, to protect the user’s location when interacting with some location-based services. The LPPMs [8], [11], [28], [29], [30], [31] generally use some obfuscation methods, like spatial cloaking, cell merging, location precision reduction or dummy cells, to manipulate the probability distribution of users’ locations. As differential privacy becomes a standard for privacy protection, [8] proposed a Geo-Indistinguishability notion based on differential privacy and a planar Laplace mechanism to achieve it. Xiao et al. [11], [20] studied how to protect location privacy under temporal correlations with an optimal differentially private mechanism. Rodriguez-Carrion et al. [32] also studied the effect of temporal dependencies on entropy-based location privacy metrics. They proposed a new privacy metrics entropy rate and perturbative mechanisms based on it, which can be an alternative LPPM in our framework for protecting spatiotemporal event privacy. Several studies [33], [34], [35] tried to achieve an optimal trade-off between the utility of applications and the privacy guarantee in the LPPMs. Overall, above works all focused on the mechanisms of location privacy, which can be used in our framework as given LPPMs. Whereas we study a new problem of spatiotemporal event privacy.

6.2 Inferences on Location
Various inference attacks can be carried out based on location information and external information such as moving patterns. In the aggregated setting, recent works have studied location or trajectory recovery attacks from aggregated location data [6], [36] or proximity query results from location data [4]. We mainly discuss the individual setting that is closely related to our work. Studies [33], [37], [38] investigated the question of how to formally quantify the privacy of existing LPPMs, given an adversary who can model users’ mobility using a Markov process learned from population, which is commonly used for modeling user mobility pattern. Liao et al. [39] used a hierarchical Markov model to learn and infer a user’s trajectory based on the places and temporal patterns they visited. Qiao et al. [40] used the Continuous Time Bayesian Networks to predict uncertain
A new type of privacy goal called spatiotemporal event privacy is defined. We formally define spatiotemporal events and design a privacy metrics extending the notion of differential privacy. We proposed PriSTE, a framework integrating an LPPM for protecting the spatiotemporal event privacy. An interesting direction is to find an optimal way for achieving both location privacy and spatiotemporal event privacy. Another question is how we can design a generic mechanism for spatiotemporal event privacy.

ACKNOWLEDGMENTS
This work was supported by JSPS KAKENHI Grant Number 17H06099, 18H04093, 19K20269, Microsoft Research Asia (CORE16), US National Science Foundation (NSF) under Grant No. CNS-1618932 and the AFOSR DDDAS program under grant No. FA9550-12-0240. This paper is extended from [1], Yang Cao and Yonghui Xiao contributed equally to this work.

REFERENCES
Yang Cao received the BS degree from the School of Software Engineering, Northwestern Polytechnical University, Xi’an, China, in 2008, and the MS and PhD degrees from the Graduate School of Informatics, Kyoto University, Kyoto, Japan, in 2014 and 2017, respectively. He is an assistant professor with the Department of Social Informatics, Kyoto University. His research interests include privacy preserving data analysis, data trading, and distributed computing.

Yonghui Xiao received the three BS degrees from Xi’an Jiaotong University, Xi’an, China, in 2005, the MS degree from Tsinghua University, Beijing, China, in 2011 after spending two years working in industry, and the PhD degree from the Department of Math and Computer Science, Emory University, Atlanta, Georgia, in 2017. He is currently a senior software engineer with Google working on machine learning and cloud security.

Li Xiong received the BS degree from the University of Science and Technology of China, Hefei, China, the MS degree from Johns Hopkins University, Baltimore, Maryland, and the PhD degree from the Georgia Institute of Technology, Atlanta, Georgia, all in computer science. She is a Winship distinguished research professor of computer science (and biomedical informatics) with Emory University. She and her research group, Assured Information Management and Sharing (AIMS), conduct research that addresses both fundamental and applied questions at the interface of data privacy and security, spatiotemporal data management, and health informatics.