Efficient and Private Federated Trajectory Matching

Yuxiang Wang, Yuxiang Zeng, Shuyuan Li, Yuanyuan Zhang, Zimu Zhou, *Member, IEEE*, Yongxin Tong, *Member, IEEE*

***Abstract*—Federated Trajectory Matching (FTM) is gaining increasing importance in big trajectory data analytics, supporting diverse applications such as public health, law enforcement, and emergency response. FTM retrieves trajectories that match with a query trajectory from a large-scale trajectory database, while safeguarding the privacy of trajectories in both the query and the database. A naive solution to FTM is to process the query through Secure Multi-party Computation (SMC) across the entire database, which is inherently secure yet inevitably slow due to the massive secure operations. A promising acceleration strategy is to filter irrelevant trajectories from the database based on the query, thus reducing the SMC operations. However, a key challenge is how to publish the query in a way that both preserves privacy and enables efficient trajectory filtering. In this paper, we design GIST, a novel framework for efficient Federated Trajectory Matching. GIST is grounded in Geo-Indistinguishability, a privacy criterion dedicated to locations. It employs a new privacy mechanism for the query that facilitates efficient trajectory filtering. We theoretically prove the privacy guarantee of the mechanism and the accuracy of the filtering strategy of GIST. Extensive evaluations on five real datasets show that GIST is significantly faster and incurs up to 2 orders of magnitude lower communication cost than the state-of-the-arts.**

***Index Terms*—trajectory matching, data federation, location privacy**

I. Introduction

The emergence of big trajectory data, powered by diverse sensors such as GPS, surveillance cameras, and proximity sensors, has revolutionized our ability to capture and analyze movement patterns. This data, often collected by various entities ranging from tech companies to government agencies, offers a multifaceted view of human mobility and urban activities. However, the distributed nature of data ownership, coupled with the inherent sensitivity of trajectory data [1], [2], necessitates paradigms that respect privacy constraints while enabling effective analysis across different data owners.

Of our particular interest is Federated Trajectory Matching (FTM), a primitive in privacy-preserving trajectory analysis across distributed data owners. FTM retrieves trajectories in a large-scale private dataset, held by a distinct data owner, that match with a query trajectory. Importantly, this query process should safeguard two categories of trajectory privacy: *(i)* the exact spatiotemporal information in the query trajectory; and *(ii)* any trajectories in the database other than the query result. We illustrate the use cases of Federated Trajectory Matching query via the following real-world applications.

Example 1 (Tracing Infections in Epidemics [3]): During a contagious disease outbreak, health officials often face the task of tracing infection paths from a limited location history. They may turn to the trajectory database of the map service providers. However, the raw location history is confidential, as its disclosure might induce panic. Likewise, it is crucial for the map service providers to prevent the trace of the uninfected individual from leakage.

Example 2 (Tracking Criminal Suspects [4]): The police often locate a criminal suspect by analyzing trajectory data from surveillance cameras or witnesses. They can improve the tracking by collaborating with map service providers via the dense GPS trajectories. However, regulations strictly limit the sharing of sensitive trajectory data with law enforcement [5]–[7], and the police are equally constrained from providing the raw query trajectory to map service providers, as these may contain confidential information.

A central challenge in FTM is to attain high query efficiency over large-scale data. While Secure Multi-party Computation (SMC) effectively ensures privacy, it falls short in terms of efficiency. As our empirical study (Sec. V-B) shows, processing a single FTM query on a database containing 3.2 million trajectories using SMC techniques [8], [9] can take as long as 89 hours. Such processing times are impractical in situations where swift responses are critical, such as in managing public health emergencies or conducting criminal investigations.

Considering that only a small portion of trajectories in the database matches the query, a natural acceleration strategy is to filter trajectories unlikely to match the query and reduce the number of SMC operations, avoiding scanning the whole trajectory database. Realizing this strategy, however, is non-trivial. The two main challenges are: *(i)* designing a privacy mechanism that enables accurate trajectory filtering, and *(ii)* developing an effective filtering scheme that operates on perturbed query trajectories. Although several privacy mechanisms in spatial/trajectory data have been proposed [10]–[12], they are not primarily designed for trajectory filtering, which can incur high retention rate (See Sec. V-D).

To this end, we present GIST (Geo-I accelerated SMC based method for federated Trajectory matching), an efficient framework for FTM queries. GIST is grounded in Geo-Indistinguishability [10], a recognized differential privacy standard for location data. It incorporates novel privacy mechanisms and trajectory filtering strategies tailored to FTM. Specifically, the query trajectory is perturbed using a newly devised Bounded Planar Laplace (BPL) mechanism and then shared with the data owner at a grid level, allowing the data

Y. Wang, Y. Zeng, S. Li, and Y. Tong are with the State Key Laboratory of Software Development Environment and Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, School of Computer Science, Beihang University, China. E-mail: {yuxiangwang, yxzeng, lishuyuan, yxtong}@buaa.edu.cn.

Y. Zhang is with the North China Institute of Computing Technology, Beijing, China. E-mail: lucidityuan@gmail.com.

Z. Zhou is with the School of Data Science, City University of Hong Kong, Hong Kong SAR, China. E-mail: zimuzhou@cityu.edu.cn.

(Corresponding authors: Yuanyuan Zhang and Yongxin Tong.)

owner to conduct effective trajectory filtering. The trade-off between the trajetory filtering granularity and the privacy parameters is analyzed theoretically. Moreover, we devise a data partition scheme along with a reference trajectory based pruning strategy to further improve the query efficiency.

Our major contributions are summarized as follows:

- We define Federated Trajectory Matching (FTM), an emerging problem in privacy-aware big trajectory data analysis that has various real-world applications.
- We propose GIST, a framework to accelerate FTM on large-scale data while accounting for privacy. The key is to reduce the number of secure operations for trajectory matching via Geo-Indistinguishability. To the best of our knowledge, this is the first work that adopts this strategy to trajectory matching.
- We develop a novel grid-level query trajectory publishing method which ensures both privacy guarantee and trajectory filtering efficiency. We theoretically analyze the trade-off between privacy level and filtering efficiency.
- Extensive experiments on five real datasets show that our solution outperforms the state-of-the-arts [8], [10]–[12] by a large margin.

The rest of paper is organized as follows. In Sec. II, we present the problem definition and related concepts. Then, we introduce the overall framework in Sec. III and elaborate on the technical details in Sec. IV. Finally, we conduct the experimental evaluation in Sec. V, review existing studies in Sec. VI, and conclude in Sec. VII.

II. Preliminaries

This section presents the problem definition (Sec. II-A) and some prerequisites on Geo-Indistinguishability (Sec. II-B).

A. Problem Definition

Definition 1 (Point [13]): Each point p is denoted by a timestamp $p.ts$ and the geo-location $p.loc$ at this timestamp.

For any two points p, q, the Euclidean distance function $\mathsf{d}(p, q)$ computes the distance between p and q.

Definition 2 (Trajectory [13]): A trajectory T is defined as a sequence of $|T|$ points, *i.e.*, $T = \langle p_1, p_2, \ldots, p_{|T|}\rangle$.

In practice, points in a trajectory can be simplified as a piecewise linear function of the timestamp [14] and each piece of the function is defined as a segment in the following.

Definition 3 (Segment [14]): A segment $s = \langle o, d\rangle$ is represented by a pair of points. The points o and d represent the origin and destination points of the segment, and satisfy the timestamp condition $o.ts \leq d.ts$. The movement between o and d is considered as linear.

Linear interpolation can be employed to derive the location of a segment s at any timestamp [14]. Specifically, we calculate the velocity of the segment as $\overline{v} = \frac{d.loc - o.loc}{d.ts - o.ts}$ and estimate the location of s at timestamp ts' as $loc_s(ts') = o.loc + (ts' - o.ts) \cdot \overline{v}$. In addition, the location of a trajectory T at timestamp ts' is computed as $loc_T(ts') = loc_s(ts')$, where s is a segment in T and satisfies $ts' \in [s.o.ts, s.d.ts]$.

Example 3: Consider trajectory $T_0 = \langle p_1, p_2, p_3, p_4\rangle$ in Fig. 1, where $p_1.loc = (2, 1)$, $p_1.ts = 0$, $p_2.loc = (1, 2)$, $p_2.ts = 2$, $p_3.loc = (4, 5)$, $p_3.ts = 5$, $p_4.loc = (6, 1)$, $p_4.ts = 7$. Trajectory T_0 can be seen as a sequence of segments $\langle s_1, s_2, s_3\rangle$, where $s_1 = \langle p_1, p_2\rangle$, $s_2 = \langle p_2, p_3\rangle$, $s_3 = \langle p_3, p_4\rangle$. We apply linear interpolation to segment s_2 to estimate the location of T_0 at timestamp 4. Namely, $\overline{v} = \frac{((4,5)-(1,2))}{5-2} = (1, 1)$, $loc_{T_0}(4) = (1, 2) + (4 - 2) \cdot \overline{v} = (3, 4)$.

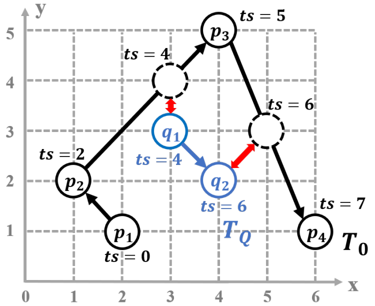


Fig. 1: The example of trajectory match.

Definition 4 (Trajectory Matching): Given a distance threshold τ and a trajectory T_Q, a trajectory T_i is considered to be matched with T_Q, denoted as $\mathsf{match}_\tau(T_i, T_Q) = \mathsf{true}$, if for every point $q \in T_Q$:

$$\mathsf{d}(q.loc,\ loc_{T_i}(q.ts)) \leq \tau \tag{1}$$

where $loc_{T_i}(q.ts)$ represents the location of trajectory T_i with the same timestamp as point q.

In practice, trajectory matching requires that each location in T_Q has a corresponding location in T_i that is sufficiently close (*i.e.*, $\leq \tau$). The definition is akin to the frequently-used spatiotemporal distance measure STED [15], [16], and the requirement that each location in the query trajectory be matched makes it more suitable for our application scenario.

Example 4: Consider trajectory T_0 and T_Q in Fig. 1. The query trajectory $T_Q = \langle q_1, q_2\rangle$, where $q_1.loc = (3, 3), q_1.ts = 4$, $q_2.loc = (4, 2), q_2.ts = 6$. We set the distance threshold $\tau = 1.5$. According to the definition, we examine timestamps 4 and 6, computing $loc_{T_0}(4) = (3, 4)$ and $loc_{T_0}(6) = (5, 3)$. Because $\mathsf{d}(q_1.loc, loc_{T_0}(4)) = \sqrt{0^2 + 1^2} = 1 < 1.5$ and $\mathsf{d}(q_2.loc, loc_{T_0}(6)) = \sqrt{1^2 + 1^2} = \sqrt{2} < 1.5$, we conclude that T_0 can be matched with T_Q.

Definition 5 (Trajectory Data Federation): The trajectory data federation comprises one or more data owners, each autonomously managing their local trajectory data. When a user submits a query, data owners and the query user collaborate to execute the queries [17]–[19]. Due to the high sensitivity of trajectory data [1], [2], trajectories other than the query result cannot be leaked to the query user or other data owners during the query execution.

Definition 6 (Federated Trajectory Matching (FTM)): Given a trajectory data federation TD containing a large amount of trajectories, a query trajectory T_Q, and a distance threshold τ, the query $\mathsf{FTM}(TD, T_Q)$ aims to securely retrieve all trajectories in TD that match with T_Q:

$$\mathsf{FTM}(TD, T_Q) = \{T_i | T_i \in TD \wedge \mathsf{match}_\tau(T_i, T_Q) = \mathsf{true}\}$$

It is required that the FTM query procedure prevents the leakage of spatiotemporal information about points in T_Q

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3424411

to the data owner. Besides, information about unmatched trajectories in TD cannot be disclosed to the query user.

B. Geo-Indistinguishability for Protecting Location Privacy

Geo-Indistinguishability (Geo-I) [10] extends the de facto standard notion of privacy protection, *i.e.*, ϵ-differential privacy (ϵ-DP) [20], to spatial data. Geo-I is widely adopted in the location-based systems and can be utilized to safeguard privacy in FTM. A mechanism M operates as a probabilistic function, taking any location within $\mathbb{X}$ as input and mapping it into a location within $\mathbb{Y}$ as output.

Definition 7 (ϵ-Geo-Indistinguishability (ϵ-Geo-I) [10]): A mechanism M satisfies ϵ-Geo-Indistinguishability (ϵ-Geo-I) iff for all $x, x' \in \mathbb{X}$ and all $Y \subseteq \mathbb{Y}$:

$$Pr[M(x) \in Y] \leq e^{\epsilon \mathsf{d}(x,x')} Pr[M(x') \in Y] \tag{2}$$

Planar Laplace Mechanism. Geo-Indistinguishability is usually achieved by introducing planar Laplacian noise [10], which can be generated through independent sampling of the radial distance r and polar angle θ in the plane polar coordinates.

The radius r depends on the cumulative distribution function (CDF) $C_\epsilon(r)$:

$$C_\epsilon(r) = 1 - (1 + \epsilon r)e^{-\epsilon r} \tag{3}$$

To derive the radius r from a given probability p, we can use the inverse function of $p = C_\epsilon(r)$, denoted as $C_\epsilon^{-1}(p)$:

$$C_\epsilon^{-1}(p) = -\frac{1}{\epsilon}\left[W_{-1}\left(\frac{p-1}{e}\right) + 1\right] \tag{4}$$

where W_{-1} represents the Lambert W function's -1 branch.

When generating the planar Laplacian noise, we first randomly pick p from the uniform distribution within $[0, 1]$ and then obtain $r = C_\epsilon^{-1}(p)$. After that, we choose $\theta \in [0, 2\pi]$ uniformly at random, and compute the noise as $(rcos\theta, rsin\theta)$.

III. Framework Overview

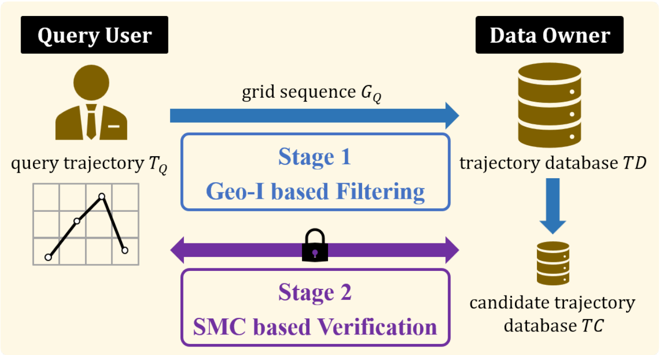


Fig. 2: Geo-I accelerated SMC based method for federated Trajectory matching (GIST).

To alleviate the high time cost of SMC in FTM query, we design a novel framework called **Geo-I accelerated SMC based method for federated Trajectory matching (GIST)**, which comprises the following two phases:

- **Geo-I based Filtering:** Initially, the user processes the query trajectory T_Q and publish it at a grid level (G_Q in Fig. 2). The procedure of trajectory publishing complies with the privacy constraint of (ϵ, δ)-Geo-I, a relaxation of standard Geo-I. Subsequently, the data owner utilizes G_Q to locally filter the database TD and obtain a reduced database TC, with the grid index accelerating the computation.
- **SMC based Verification:** Following the filtering phase, both the query user and the data owner securely verify trajectories within TC to identify all trajectories that match T_Q. We devise a data partition scheme along with a reference trajectory based pruning strategy to further improve efficiency.

TABLE I: Summary of major notations.

Notations	Description
TD, TC	trajectory database and candidate trajectory database
T_Q, τ	query trajectory and distance threshold
ϵ, δ	privacy budget and failure probability
G_Q, GI	published grid sequence and grid index
L, R	grid size and radius of noise circle
p	probability of successfully perturbing a single location
PN, rt	partition and its reference trajectory
α, m	partition parameter and maximum size of partition

We focus on the scenario with a single data owner, since the FTM in a data federation with multiple data owners can be addressed by executing the FTM query with each data owner in parallel (see Sec. V-E). The major notations used in the paper are listed in Table I.

IV. Algorithm Design

This section introduces the algorithm designs of our framework GIST from two aspects: *Geo-I based Filtering* (Sec. IV-A) and *SMC based Verification* (Sec. IV-B).

A. Geo-I based Filtering

In the following, we first introduce a location privacy definition named (ϵ, δ)-Geo-I and propose a mechanism to achieve it. Then, we provide a detailed explanation of how to filter candidate answers from TD based on the privately published query trajectory. Finally, we theoretically derive the appropriate grid size used in the filtering process.

*1) (ϵ, δ)-**Geo-Indistinguishability and Bounded Planar Laplace Mechanism**:* To pursue a promising query performance, we propose a new definition of location privacy based on ϵ-Geo-I and achieve it with a mechanism named Bounded Planar Laplace (BPL).

Motivation. In ϵ-Geo-I, the spatial range of the injected noise is usually unbounded under Planar Laplace mechanism, meaning that the original location can be perturbed to an arbitrarily distant location. In practice, this feature may lead to unexpected result: a perturbed location too far away from the original one may seriously compromise the usability (*i.e.*, query performance in our case). Thus, we aim to design a new privacy mechanism, which can not only restrict the upper bound of noise in the spatial area, but also generally follows the concept of ϵ-Geo-I.

Algorithm 1: Bounded Planar Laplace (BPL)

```
input : location $x$, privacy parameters $\epsilon, \delta$
output: perturbed location $x'$
1  Find $\Delta$ that satisfies $\Delta = (\pi\delta - \frac{1}{2}\epsilon^2)[C_\epsilon^{-1}(1-\Delta)]^2$;
2  $R \leftarrow C_\epsilon^{-1}(1-\Delta)$;
3  Choose $p \in [0,1]$ uniformly at random;
4  if $p \leq 1-\Delta$ then // planar Laplacian noise
5      $r \leftarrow C_\epsilon^{-1}(p)$;
6  else    // uniform noise in noise circle
7      Choose $r$ s.t. $r^2$ is uniformly sampled in $[0, R^2]$;
8  Choose $\theta \in [0, 2\pi]$ uniformly at random;
9  $x' \leftarrow x + (rcos\theta, rsin\theta)$;
10 return $x'$;
```

Definition of (ϵ, δ)**-Geo-I.** Motivated by the generalization of (ϵ, δ)-DP (*a.k.a.*, approximate DP) from ϵ-DP (*a.k.a.*, pure DP) [21], we introduce an approximate version of Geo-I in Definition 8, allowing a small probability δ of failing to reach ϵ-Geo-I.

Definition 8 ((ϵ, δ)-Geo-I): A mechanism M satisfies (ϵ, δ)-Geo-Indistinguishability ((ϵ, δ)-Geo-I) iff for all $x, x' \in \mathbb{X}$ and all $Y \subseteq \mathbb{Y}$:

$$Pr[M(x) \in Y] \leq e^{\epsilon \mathsf{d}(x,x')} Pr[M(x') \in Y] + \delta \qquad (5)$$

Post-processing is a crucial property for differential privacy [21]. Similarly, we can prove in Lemma 1 that this property holds true for (ϵ, δ)-Geo-I as well.

Lemma 1 (Post-processing): Give mechanism M that satisfies (ϵ, δ)-Geo-I, then for any algorithm f, the composition of M and f, *i.e.*, $f(M(\cdot))$ satisfies (ϵ, δ)-Geo-I.

Proof: We first prove the result for any deterministic function f. The lemma then follows as any randomized mapping can be decomposed into a convex combination of deterministic functions [21]. Define the output domain of f as $\mathbb{Z}$. For any $x, x' \in \mathbb{X}$, and any $Z \subseteq \mathbb{Z}$, we prove this lemma as follows:

$$\begin{aligned} Pr[f(M(x)) \in Z] &= Pr[M(x) \in Y] \\ &\leq e^{\epsilon \mathsf{d}(x,x')} Pr[M(x') \in Y] + \delta \\ &= e^{\epsilon \mathsf{d}(x,x')} Pr[f(M(x')) \in Z] + \delta \end{aligned}$$

where $Y = \{y \in \mathbb{Y} | f(y) \in Z\}$. ■

Privacy Mechanism: Bounded Planar Laplace. We devise the Bounded Planar Laplace (BPL) mechanism to achieve (ϵ, δ)-Geo-I. The *main advantage* of the BPL mechanism lies in its ability to constrain the maximum value of noise. In other words, the distance between the perturbed location and the original location cannot exceed R. For conciseness, we refer to the circle with a radius of R as the *noise circle*.

Algorithm 1 illustrates the detailed procedure of the Bounded Planar Laplace (BPL) mechanism. It begins by computing the radius of the noise circle R based on the privacy parameters ϵ and δ in lines 1-2. Then, a random p is uniformly chosen from $[0, 1]$. If p is less than a threshold $1 - \Delta$, a noise is generated using p, following the standard Planar Laplace mechanism (Equation (4)), as shown in lines 4-5. However, if p exceeds the threshold, it implies that the size of noise generated by the standard Planar Laplace mechanism exceeds R. In such cases, a uniform noise within the noise circle is selected, as demonstrated in lines 6-7. Finally, in lines 8-10, a random angle θ is chosen, and the noise $(rcos\theta, rsin\theta)$ is used to perturb x and obtain x'. The privacy guarantee of the BPL mechanism is proven in Lemma 2.

Lemma 2: The Bounded Planar Laplace (BPL) mechanism satisfies (ϵ, δ)-Geo-Indistinguishability.

Proof: We consider the probability p when perturbing the location x:

(1) If $p \leq 1 - \Delta$, a standard planar Laplacian noise is added to the location x, which satisfies ϵ-Geo-I [10].

(2) If $p > 1 - \Delta$, lines 6-7 fail to guarantee ϵ-Geo-I with probability $\frac{\epsilon^2}{2\pi} + \frac{\Delta}{\pi R^2} = \frac{\epsilon^2}{2\pi} + \frac{1}{\pi}(\pi\delta - \frac{1}{2}\epsilon^2) = \delta$, where $\frac{\epsilon^2}{2\pi}$ is the maximal probability of location when applying the planar Laplacian mechanism, and $\frac{\Delta}{\pi R^2}$ is the additional probability incurred by the uniform distribution in the noise circle.

Therefore, the BPL mechanism satisfies (ϵ, δ)-Geo-I. ■

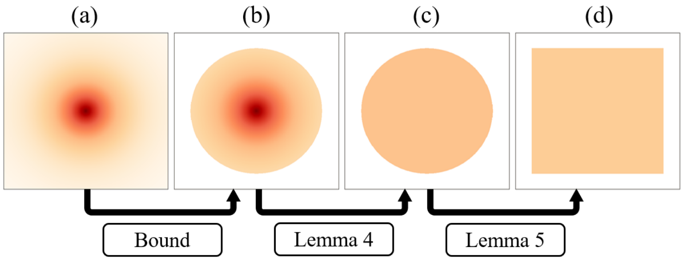


Fig. 3: The probability density function (pdf) of: (a) planar Laplacian noise; (b) bounded planar Laplacian noise; (c) uniform noise in the noise circle; (d) uniform noise in the circumscribed square of the noise circle.

Fig. 3(a) and (b) show the standard planar Laplacian noise and the bounded planar Laplacian noise, respectively. The BPL noise is rigorously constrained within a size of R, leading to improved query performance based on our experiments.

*2) **Our Filtering Algorithm**:* The filtering performs at a grid level. Both the query user and the data owner divide the spatial region into equal-sized square grids and utilize grid representations of trajectories for filtering. Initially, the query user publishes trajectory T_Q in the form of a grid sequence, denoted by G_Q. Then, the data owner utilize G_Q to locally filter TD, and employs grid index to accelerate the filtering process. The grid size should be selected carefully to ensure compliance with (ϵ, δ)-Geo-I, as discussed in Sec. IV-A3.

Query Trajectory Publishing. Based on the Bounded Planar Laplace mechanism, we develop a novel approach for publishing query trajectory which ensures (ϵ, δ)-Geo-Indistinguishability. The publishing algorithm involves two core operations: *Perturbation* and *Grid-Selection*. Perturbation obtains a perturbed location x' by adding the BPL noise to the original location, while Grid-Selection determines the grid in which x' is located.

As shown in Algorithm 2, for each location $x \in T_Q$, Perturbation is executed in line 4, followed by Grid-Selection

Algorithm 2: (ϵ, δ)-Geo-I based Filtering

```
input : trajectory database $TD$, query trajectory $T_Q$,
        privacy parameters $\epsilon, \delta$, publishing rate $\rho$
output: candidate trajectory database $TC$
1  Choose grid size $L$ according to $\epsilon, \delta, \rho$;
   // Query user's protocol;
2  $cand \leftarrow \phi$;
3  foreach location $x \in T_Q$ do
4      $x' \leftarrow \mathrm{BPL}(x, \epsilon, \delta)$;
5      $g' \leftarrow$ grid No. of $x'$;
6      if $g' =$ grid No. of $x$ then
7          Add $g'$ to $cand$;
8  Select $\lfloor \rho \cdot |T_Q| \rfloor$ grids from $cand$ into $G_Q$, and
     eliminate repetitive grids in it;
9  Send $G_Q$ to the data owner;
   // Data owner's protocol;
10 Construct the grid index $GI$ using $TD$;
11 Upon receiving $G_Q$, compute $TC = \bigcap_{g \in G_Q} GI[g]$;
12 return $TC$;
```

in line 5. Subsequently, in lines 6-7, we check whether the perturbed location x' and the original location x are located in the same grid. We add the grid to the candidate list *cand* only when they are located in the same grid, ensuring that the published grids can be precisely used for filtering. Finally, in lines 8-9, we pick $\lfloor \rho \cdot |T_Q| \rfloor$ grids from *cand* for publishing, where ρ is a ratio specified by the query user. We prove in Theorem 1 that our publishing method satisfies the privacy constraint of (ϵ, δ)-Geo-I.

Filtering Strategy. We introduce the notion of traversal grids before detailing the process of filtering the trajectory database.

Definition 9 (Traversal Grids): The traversal grids of a trajectory T under distance threshold τ, denoted as $G_\tau(T)$, contain all the grids covered by a set of circles $\{\mathsf{circle}(x, \tau) | x \in T.locs\}$. Here $\mathsf{circle}(x, \tau)$ denotes a circle centered at location x and with a radius of τ, and $T.locs$ represents all the locations in trajectory T, including intermediate locations on each segment.

Example 5: Consider trajectory $T_1 = \langle p_1, p_2, p_3 \rangle$ in Fig. 4. Segment $s_1 = \langle p_1, p_2 \rangle$ traverses grids $5, 6, 7, 11, 12$, and segment $s_2 = \langle p_2, p_3 \rangle$ traverses grids $12, 8$. Besides, $\mathsf{circle}(p_1, \tau)$ covers grid 1, and $\mathsf{circle}(p_4, \tau)$ (p_4 is a location in segment s_1) covers grid 10. Thus, the traversal grids of T_1 under τ, $G_\tau(T_1) = \{1, 5, 6, 7, 8, 10, 11, 12\}$.

Using the concept of the traversal grids, we formulate the filtering strategy as follow: the data owner filters TD by retaining only trajectories whose traversal grids encompass all the grids in G_Q. The correctness of the strategy is proven in Lemma 3.

Example 6: In Fig. 4, the query user publishes T_Q at a grid level for filtering. G_Q is a grid sequence generated by T_Q and more specifically, T_Q's subtrajectory T'_Q. When receiving G_Q, the data owner filters TD according to the traversal grids of each trajectory. In our example, T_1 is filtered out while T_2 is not, since $G_Q \nsubseteq G_\tau(T_1)$ and $G_Q \subset G_\tau(T_2)$.

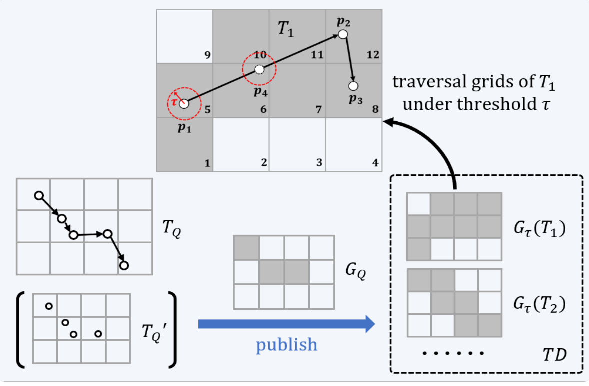


Fig. 4: An illustration of Geo-I based Filtering.

Indexing. Considering the substantial amount of trajectory data in the database TD, we introduce a grid index to accelerate our filtering strategy. The grid index GI is constructed during the offline stage. For each trajectory in TD, we calculate its traversal grids and insert all the mapping relations from grid ID to trajectory ID into GI. During the online stage, we can efficiently obtain the candidate database TC by computing the intersection of $GI[g]$ for all $g \in G_Q$. The time for index construction is excluded from the complexity analysis since it can be finished before the query execution.

Correctness of Our Filtering. The correctness of the filtering strategy is proven in Lemma 3.

Lemma 3: If the query user publishes the grid sequence G_Q using trajectory T_Q, then $G_Q \subseteq G_\tau(T_i)$ is a necessary condition for $\mathsf{match}_\tau(T_i, T_Q) = \mathsf{true}$.

Proof: Suppose there is a grid $g \in G_Q$ such that $g \notin G_\tau(T_i)$. Then the point q that generates grid g, can never be matched by any locations in T_i, even when disregarding timestamps. This implies that $\mathsf{match}_\tau(T_i, T_Q)$ should always be false in such cases, thereby completing our proof. ■

Privacy of Query Trajectory Publishing. We prove the privacy guarantee of query trajectory publishing in Theorem 1.

Theorem 1: The query trajectory publishing algorithm satisfies (ϵ, δ)-Geo-Indistinguishability.

Proof: We analyse the two core operations, Perturbation and Grid-Selection, respectively. According to Lemma 2, Perturbation satisfies (ϵ, δ)-Geo-I. Grid-Selection can be viewed as a post-processing after perturbation since the grid number is determined only based on x' and does not rely on x.

Lemma 1 has proven that post-processing does not impact the privacy guarantee. Thus, publishing location x as g' satisfies (ϵ, δ)-Geo-I. Suppose G_Q is generated by T'_Q, a subtrajectory of T_Q, then the procedure of publishing T'_Q as G_Q preserves (ϵ, δ)-Geo-I, which completes our proof. ■

After the query is published, the data owner conducts local filtering utilizing the published trajectory (line 10 in Algorithm 2). Since the filtering avoids any interaction with the query user, the Geo-I filtering also complies with the privacy constraint of (ϵ, δ)-Geo-I.

Complexity Analysis. Given that each entry in grid index GI is sorted, the time complexity of filtering with the grid index is $O(\sum_{q \in G_Q} |GI[q]|)$, where G_Q is the published grid sequence.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3424411

*3) **Selection of Grid Size:*** As indicated in Algorithm 2, the grid size is closely related to the privacy level and the query performance. Therefore, it is crucial to derive a proper filtering granularity that can achieve the specified privacy level. To this end, we theoretically analyse the *relation between the grid size and the privacy parameter* in Theorem 2.

Basic Idea. To determine the grid size, we need to establish a connection between the probability of successfully perturbing one location p, the radius of the noise circle R and the grid size L. Given R and L, we can estimate p by considering the probability that both location x and its perturbation x' fall in the same grid. Our analysis is based on the success probability in three different types of areas in a grid: *center area*, *side area*, and *corner area*, as shown in Fig. 5.

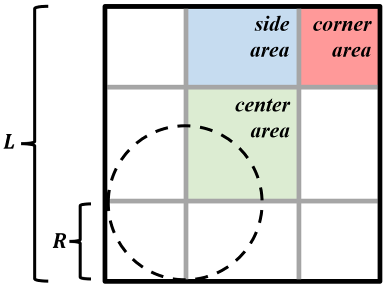


Fig. 5: Division of a grid into three types of areas. A grid is a square with a side length of L. The radius of the noise circle is denoted as R ($R \le \frac{L}{3}$).

Upper Bound for Grid Size L. We use $p_{center}, p_{side}, p_{corner}$ to denote the probability of successful perturbation in three types of area, then the following equation holds:

$$L^2 p = (L-2R)^2 p_{center} + 4R(L-2R) p_{side} + 4R^2 p_{corner} \quad (6)$$

We note that $p_{center} = 1$, as for any location in the center area, the distance from the location to the grid boundary consistently exceeds R, the maximum size of the BPL noise.

We then proceed to analyse p_{side} and p_{corner}. Considering the complexity of the planar Laplace distribution, we use an approximation to simplify the analysis. Specifically, we replace the BPL noise (Fig. 3(b)) with the uniform noise in the circumscribed square of the noise circle (Fig. 3(d)). The feasibility of this replacement will be discussed in Lemma 4 and Lemma 5.

Theorem 2: Suppose p represents the success probability of perturbing a single location, then a grid size $L = \frac{R}{2(1-\sqrt{p_0})}$ can ensure $p \ge p_0$.

Proof: According to Lemma 4 and Lemma 5, replacing the BPL noise with the uniform noise in the circumscribed square of the noise circle reduces the probability of successful perturbation. Thus, we can derive the lower bounds for p_{side} and p_{corner}:

(1) Consider the blue point in Fig. 6, p_{side} satisfies:

$$p_{side} \ge \int_0^R \frac{1}{4R^2} \cdot 2R(2R-z)dz = \frac{3}{4}$$

(2) Consider the red point in Fig. 6, p_{corner} satisfies:

$$p_{corner} \ge \int_0^R \int_0^R \frac{1}{4R^2}[(2R-x)(2R-y)]\ dxdy = \frac{9}{16}$$

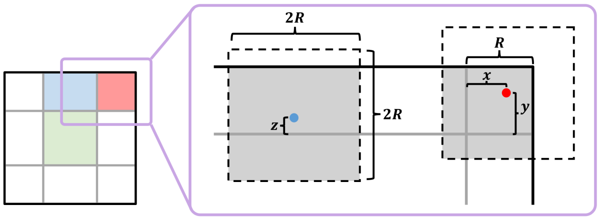


Fig. 6: Using the approximation to obtain lower bounds for p_{side} and p_{corner}. The noise is uniformly sampled in the circumscribed square of the noise circle, which is the square with a side length of $2R$. The noise in the grey area ensures the successful perturbation.

Substituting $p_{center} = 1, p_{side} \ge \frac{3}{4}, p_{corner} \ge \frac{9}{16}$ into Equation (6), we obtain:

$$L^2 p \ge (L-2R)^2 + \frac{3}{4} \cdot 4R(L-2R) + \frac{9}{16} \cdot 4R^2$$
$$\Rightarrow p \ge (1 - \frac{R}{2L})^2$$

Thus, $L = \frac{R}{2(1-\sqrt{p_0})}$ can ensure that $p \ge p_0$. ■

Lemma 4 and Lemma 5 prove the feasibility of the replacement by leveraging the uniform noise in the noise circle (Fig. 3 (c)) as an intermediate.

Lemma 4: Replacing the bounded planar Laplacian noise with the uniform noise in the noise circle reduces the probability of successful perturbation.

Proof: We denote the probability of generating a BPL noise of size x as $g(x)$, and the probability of generating a uniform noise of size x as $u(x)$. Then we have:

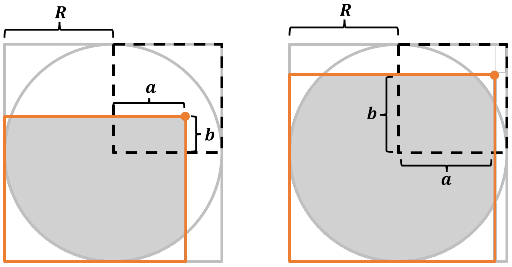
$$\int_0^R u(x)xdx = \int_0^R g(x)xdx = 1 \quad (7)$$

We can observe that $u(x) \equiv \frac{2}{R^2}$, and $g(x)$ is monotonically decreasing in $[0, R]$. Then we use $f(x)$ to represent the average probability of successful perturbation when the noise size is x. This function is monotonically decreasing in $[0, R]$, as a larger noise reduces the success probability. Based on the monotonicity of $f(x)$ and $g(x)$, we observe that for all $x, y \in [0, R], x \ne y$, $[f(x)-f(y)][g(x)-g(y)] > 0$, indicating that $[f(x)-f(y)][g(x)-g(y)]xy \ge 0$, hence:

$$0 \le \int_0^R \int_0^R [f(x)-f(y)][g(x)-g(y)]xydxdy$$
$$= \int_0^R f(x)g(x)xdx \int_0^R ydy + \int_0^R f(y)g(y)ydy \int_0^R xdx$$
$$- \int_0^R \int_0^R f(x)g(y)xydxdy - \int_0^R \int_0^R f(y)g(x)xydxdy$$

According to the symmetry under double integral interchange, $\int_0^R \int_0^R f(x)g(y)xydxdy = \int_0^R \int_0^R f(y)g(x)xydxdy$, thus,

$$0 \le \frac{R^2}{2} \cdot \int_0^R f(x)g(x)xdx + \frac{R^2}{2} \cdot \int_0^R f(y)g(y)ydy$$
$$- 2 \cdot \int_0^R \int_0^R f(x)g(y)xydxdy$$



(a) Inside Circle (b) Outside Circle

Fig. 7: The orange point denotes the upper right point of a grid, and is located within the dash box. $C_\cap$ denotes the size of the grey area (intersection between the noise circle and the orange rectangle), and C denotes the size of the noise circle. $S_\cap$ denotes the size of the orange rectangle, and S denotes the size of the large square.

$$\Rightarrow \frac{R^2}{2} \cdot \int_0^R f(x)g(x)xdx \geq \int_0^R \int_0^R f(x)g(y)xydxdy$$
$$= \int_0^R f(x)xdx \int_0^R g(y)ydy$$

since Equation (7) indicates that $\int_0^R g(y)ydy = 1$ and $u(x) \equiv \frac{2}{R^2}$, we can obtain:

$$\int_0^R f(x)g(x)xdx \geq \frac{2}{R^2} \int_0^R f(x)xdx = \int_0^R f(x)u(x)xdx$$

where $\int_0^R f(x)u(x)xdx$ is the success probability using the uniform noise, and $\int_0^R f(x)g(x)xdx$ is the success probability using the BPL noise, thus Lemma 4 holds true. ■

Lemma 5: Replacing the uniform noise in the noise circle with the uniform noise in the noise circle's circumscribed square reduces the probability of successful perturbation.

Proof: As shown in Fig. 7, we use the orange point and the grey circle to represent the relative location between the grid and the noise circle. Then we can prove Lemma 5 by deriving the following inequation:

$$\frac{C_\cap}{C} \geq \frac{S_\cap}{S} \tag{8}$$

where $\frac{C_\cap}{C}$ is the successful probability using the noise in the noise circle, and $\frac{S_\cap}{S}$ is the successful probability using the noise in the circumscribed square.

(1) If the random location lies inside the noise circle (*i.e.*, $a^2 + b^2 \leq R^2$), as shown in Fig. 7(a):

$$C_\cap = \frac{\pi R^2}{4} + ab + \frac{1}{2}(a\sqrt{R^2 - a^2} + b\sqrt{R^2 - b^2})$$
$$+ \frac{R^2}{2}(arcsin\ \frac{a}{R} + arcsin\frac{b}{R})$$

(2) If the random location lies outside the noise circle (*i.e.*, $a^2 + b^2 > R^2$), as shown in Fig. 7(b):

$$C_\cap = a\sqrt{R^2 - a^2} + b\sqrt{R^2 - b^2} + R^2(arcsin\frac{a}{R} + arcsin\frac{b}{R})$$

Besides, we have $S_\cap = (R+a)(R+b)$, $C = \pi R^2$, $S = 4R^2$. It can be confirmed that for all $a, b \in [0, R]$, Equation (8) holds true, which completes our proof. ■

B. SMC based Verification

After filtering, we reduce the search space to a smaller candidate database TC, where each trajectory traverses all grids in the published grid sequence G_Q. However, as SMC operations are typically slower than their plaintext counterparts [9], performing SMC over the potentially large TC is time-consuming. Thus, we introduce a data partition scheme along with a reference trajectory based pruning strategy to further improve efficiency.

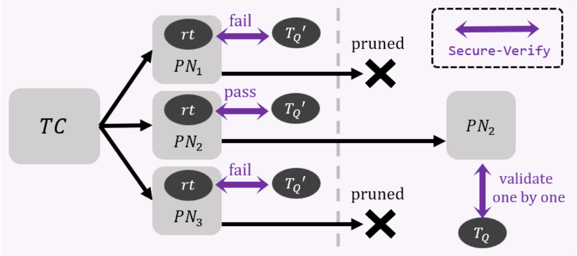


Fig. 8: An illustration of SMC based Verification.

Basic Idea. The idea of SMC based verification is illustrated in Fig. 8. We devide the candidate database TC into multiple data partitions PN_i based on the spatiotemporal characteristics of trajectories. For each partition PN_i, we generate a special trajectory, termed the *reference trajectory* rt, which encapsulates the spatiotemporal features of all trajectories in PN_i. We then apply Lemma 6 to prune partitions where none of the trajectories can match T_Q, avoiding the need to validate each trajectory in TC through SMCs.

Data Partition. Based on the spatiotemporal information of trajectories, we divide the database TC into multiple partitions, each containing no more than m trajectories. We consider trajectories that traverse a grid in an approximate time range as similar and group them into the same partition. Specifically, the data partition is performed as follows. We begin by grouping all trajectories in TC into a single partition. Subsequently, this partition is split recursively until each resulting partition reaches a size smaller than m. The splitting operation can divide one partition into two, while ensuring that trajectories within each resulting partition exhibit relatively similar spatiotemporal characteristics.

The splitting operation for PN is based on the timespan of the trajectories within it. We select the grid with the longest timespan as the splitting criterion, denoted as g_{split}. This means trajectories in PN are separated into different partitions based on when they traverse grid g_{split}. Then the splitting value, v_{split}, is set to be the median of the ending times for all trajectories in partition PN. As a result, PN can be divided into two partitions of equal sizes or sizes differing by 1.

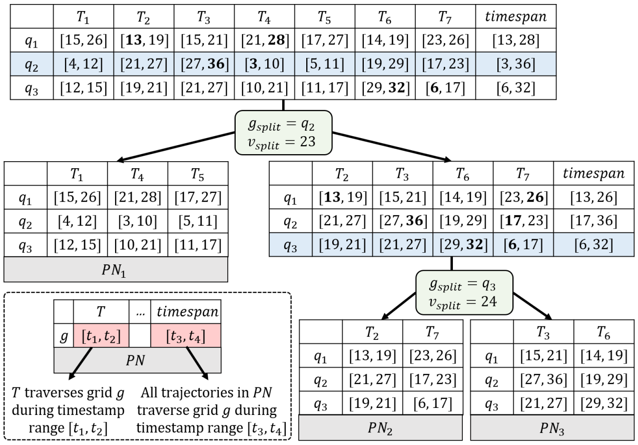
Example 7: As shown in Fig. 9, we set $m = 3$ and start with the partition with 7 trajectories. At first, we identify q_2 as the splitting criteria, since it has the longest timespan of $[3, 36]$. Then we determine the splitting value as 23, which is the median in list $[12, 27, 36, 10, 11, 29, 23]$. After that, we split the whole partition into two partitions, PN_1 with 3 trajectories, and the other with 4 trajectories, which is then split into PN_2 and PN_3 using q_3 as the splitting criterion.

Algorithm 3: SMC based Verification

```
input : candidate trajectory database $TC$,
        query trajectory $T_Q$, distance threshold $\tau$,
        grid size $L$
output: all trajectories that matches $T_Q$
1  $result \leftarrow \phi$;
2  Partition $TC$ with $m \leftarrow \lfloor \alpha\sqrt{|TC|} \rfloor$ to obtain $PNs$;
3  foreach partition $PN \in PNs$ do
4      Generate $rt$ as the reference trajectory of $PN$;
       // Pruning
5      $match_{rt} \leftarrow$ Secure-Verify($rt, T'_Q, \tau + \sqrt{2}L$);
6      if $match_{rt}$ is false then
7          continue;
       // Final Validation
8      foreach trajectory $t \in PN$ do
9          $match_t \leftarrow$ Secure-Verify($t, T_Q, \tau$);
10         if $match_t$ is true then
11             Add $t$ to $result$;
12 return $result$;
13 Function Secure-Verify($T$, $T_Q$, $\tau$):
14     $N \leftarrow 0$;
15     foreach point $q \in T_Q$ do
16         $match_q \leftarrow 0$;
17         foreach segment $s \in T$ do
18             Compute $loc_s(ts)$, which derives the
               location of $s$ at timestamp $ts$;
19             $dis \leftarrow \mathsf{d}(q.loc,\ loc_s(q.ts))$;
20             if $dis \leq \tau$ and $q.ts \in [s.o.ts, s.d.ts]$ then
21                 $match_q \leftarrow 1$;
22         $N \leftarrow N + match_q$;
23     $match \leftarrow N = |T_Q|$;
24     return $match$;
```

Pruning with Reference Trajectory. We use the reference trajectory within each partition for pruning. The reference trajectory of partition PN is generated by scanning all trajectory points of PN that located within grid g for each $g \in G_Q$. The point with the smallest timestamp is selected as the original point o, and the one with the largest timestamp is selected as the destination point d. The segment formed by o and d is added to the reference trajectory rt. This process is repeated for each grid $g \in G_Q$, resulting in a reference trajectory with $|G_Q|$ segments. Then rt can be used for pruning according to Lemma 6.

Algorithm Details. The procedure of SMC based verification is presented in Algorithm 3. In line 2, data partition is performed on the candidate trajectories TC, with the partition size limited to $\lfloor \alpha\sqrt{|TC|} \rfloor$. The parameter α is adjustable and discussed in Section V-D. Subsequently, in line 4, a reference trajectory rt is generated for each partition, and lines 5-7 utilize it for pruning based on Lemma 6. If rt does not match TQ' under the threshold of $\sqrt{2}L + \tau$, all trajectories in its partition are pruned. Following pruning, every trajectory in the remaining partitions undergoes final validation via Secure-Verify to ascertain whether they match T_Q, as shown in lines 8-11.



	T_1	T_2	T_3	T_4	T_5	T_6	T_7	*timespan*
q_1	[15, 26]	[**13**, 19]	[15, 21]	[21, **28**]	[17, 27]	[14, 19]	[23, 26]	[13, 28]
q_2	[4, 12]	[21, 27]	[27, **36**]	[**3**, 10]	[5, 11]	[19, 29]	[17, 23]	[3, 36]
q_3	[12, 15]	[19, 21]	[21, 27]	[10, 21]	[11, 17]	[29, **32**]	[**6**, 17]	[6, 32]

$g_{split} = q_2$
$v_{split} = 23$

	T_1	T_4	T_5
q_1	[15, 26]	[21, 28]	[17, 27]
q_2	[4, 12]	[3, 10]	[5, 11]
q_3	[12, 15]	[10, 21]	[11, 17]
PN_1			

	T_2	T_3	T_6	T_7	*timespan*
q_1	[**13**, 19]	[15, 21]	[14, 19]	[23, **26**]	[13, 26]
q_2	[21, 27]	[27, **36**]	[19, 29]	[**17**, 23]	[17, 36]
q_3	[19, 21]	[21, 27]	[29, **32**]	[**6**, 17]	[6, 32]

$g_{split} = q_3$
$v_{split} = 24$

	T	...	*timespan*
g	$[t_1, t_2]$		$[t_3, t_4]$
PN			

T traverses grid g during timestamp range $[t_1, t_2]$
All trajectories in PN traverse grid g during timestamp range $[t_3, t_4]$

	T_2	T_7
q_1	[13, 19]	[23, 26]
q_2	[21, 27]	[17, 23]
q_3	[19, 21]	[6, 17]
PN_2		

	T_3	T_6
q_1	[15, 21]	[14, 19]
q_2	[27, 36]	[19, 29]
q_3	[21, 27]	[29, 32]
PN_3		

Fig. 9: Divide database $TC = \{T_1, ..., T_7\}$ into 3 partitions. All the trajectories in TC traverse grids q_1, q_2 and q_3.

The Secure-Verify function in Algorithm 3 securely verify whether a trajectory T matches T_Q based on Definition 4. In lines 15-17, each point $q \in T_Q$ is iterated through to check if it can be matched by at least one segment in T. For each segment s, the data owner locally derives a function $loc_s(ts)$ for estimating the location of s at timestamp ts (line 18). In line 19, the Euclidean distance dis between $q.loc$ and its corresponding point in s is securely computed. If dis is less than τ and the sequential order of timestamps is satisfied, $match_q$ is set to 1, indicating that q can be matched by s (lines 20-21). If at least one segment matches q, the total matching number N increases by 1 (line 22). Lines 23-24 securely compare the total matching number with the length of T_Q and output the result indicating whether T matches T_Q or not. The code in lines 19-23 need to be implemented with two-party SMC protocols (*e.g.*, by using Obliv-C [22]).

Correctness of Our Pruning. The correctness of pruning in lines 5-7 of Algorithm 3 is proven in Lemma 6.

Lemma 6: If the reference trajectory rt of partition PN fails to match T'_Q under a threshold of $\tau + \sqrt{2}L$ ($L > \tau$), then no trajectory in PN can match T_Q under a threshold of τ. Here, T'_Q denotes the subtrajectory of T_Q that generates G_Q, and L denotes the grid size.

Proof: We prove Lemma 6 by demonstrating that if a point q in T'_Q cannot be matched by any segments in rt under a threshold of $\sqrt{2}L + \tau$, it cannot either be matched by any trajectories in PN under a threshold of τ.

Consider the case where the distance from q to the grid boundary is always larger than τ, implying that q can only be matched by locations within the same grid. Since $\sqrt{2}L$ is the longest distance between two locations within a grid, none of the segments in rt traverse q's grid at timestamp $q.ts$. Consequently, no trajectory in PN traverses q's grid at timestamp $q.ts$, indicating that q cannot be matched by any trajectory in PN. We can extend this conclusion to the general

TABLE II: Real datasets.

Dataset	Geolife	Dazhong	Xi'an	Chengdu	Multi-Company
$\|TD\|$	11k	700k	3200k	6000k	2400k
Size	0.3G	1.6G	10.9G	16.7G	38.7G

TABLE III: Parameter settings.

Parameter	Setting
Sampling Rate	5%, 10%, 20%, 40%
Trajectory Scalability $\|TD\|$	1500k, 3000k, 4500k, 6000k
Privacy Budget ϵ	0.01, 0.02, 0.03, 0.04, 0.05
Partition Parameter α	0.25, 0.5, 1, 2, 4
#(Data Owner)	1, 2, 3, 4, 5

case by raising the threshold from $\sqrt{2}L$ to $\sqrt{2}L + \tau$, since $L > \tau$ indicates that q can only be matched by the point from the adjacent grid, thus completing our proof. ■

Security of SMC Based Verification. In SMC based verification, the number of reference trajectories and whether they pass the pruning stage, along with the lengths of trajectories in the remaining partitions are disclosed to facilitate the execution of `Secure-Verify`. Apart from this, all other information regarding trajectories in TC and query trajectory T_Q are thoroughly protected by SMC in the semi-honest model.

Complexity Analysis. The complexity of pruning and final validation is $O(|T'_Q| \cdot |G_Q| \cdot \frac{|TC|}{m} + |T_Q| \cdot |T| \cdot n_r m)$, where n_r is the number of partitions surviving the pruning. According to the experiments on real-world datasets, n_r can be regarded as a constant. Besides, $|T_Q|$, $|T'_Q|$, $|G_Q|$ and $|T|$ are constants related to the trajectory length. Therefore, we choose $m = \Theta(\sqrt{|TC|})$ (*i.e.*, $m = \lfloor \alpha\sqrt{|TC|} \rfloor$) in line 2 of Algorithm 3 to achieve optimal complexity.

V. Experimental Study

This section presents our experimental evaluation. We first introduce the experiment setup (Sec. V-A). Then, we present the performance on real datasets (Sec. V-B), scalability tests (Sec. V-C), ablation studies (Sec. V-D), and extension to multiple data owners (Sec. V-E).

A. Experiment Setup

Datasets. We use five real-world trajectory datasets.

- **Geolife [23].** It contains daily trajectories of individuals collected by MSRA from April 2007 to August 2012.
- **Dazhong [24].** It contains trajectories of 13,013 cars collected by SAIC Volkswagen [25] in April 2016 and May 2016.
- **Xi'an & Chengdu [26].** They are trajectory datasets published by Didi Chuxing's ride-hailing services in Xi'an and Chengdu, respectively, during October 2016.
- **Multi-Company [18].** It is a shared trajectory dataset from 5 taxi companies in Beijing (*e.g.*, JinYinJian [27]). Each company can be naturally regarded as a data owner of its collected trajectories.

Baselines. We compare our framework GIST with following solutions:

- **STSC-ext [8].** It uses both addictively homomorphic encryption and secure multi-party computation to calculate the similarity between trajectories. When extending this method to FTM, we assume that timestamps of points in T_Q are published first.
- **PL-filter [10].** It uses Planar Laplace mechanism to publish points and guarantee Geo-I. We extend the method by using the trajectory published by Planar Laplace mechanism for filtering.
- **NGram-filter [11].** It uses exponential mechanism to publish the trajectory under local differential privacy. Then the published trajectory is used to filter the trajectory database.
- **ATP-filter [12].** It uses direction information to perturb the query trajectory so that the published trajectory satisfies ϵ-LDP. Then the published trajectory is utilized for filtering.

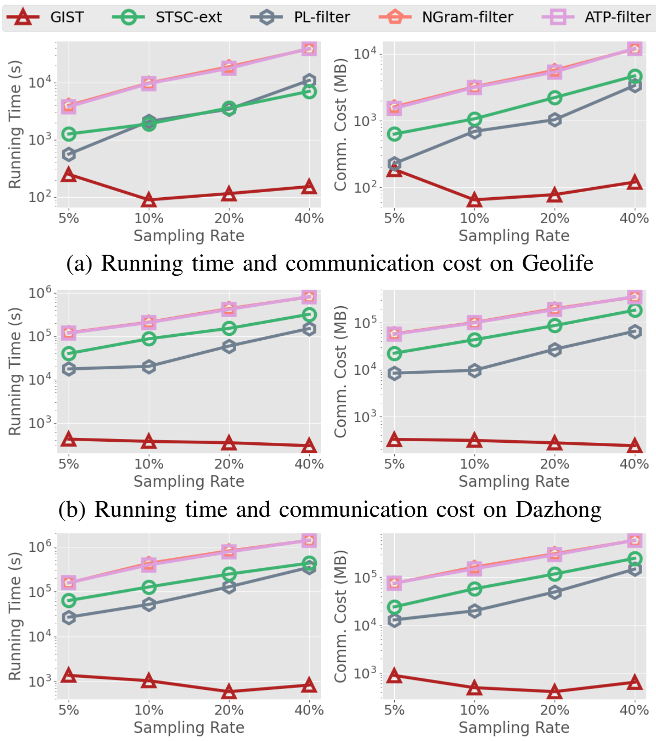
The last three baselines, namely PL-filter, NGram-filter and ATP-filter, follow a same procedure as GIST. At first, the query trajectory T_Q is published as T'_Q, using the corresponding privacy mechanism, and the published trajectory is used for filtering. Then each trajectory in the filtered database undergoes secure verification one by one. To guarantee the accuracy of the filtering process, we use the safe threshold for these methods. The safe threshold $\mathcal{T}$ ensures that if the distance between trajectory T and T'_Q is larger than $\mathcal{T}$, then T can be ruled out safely. Specifically, $\mathcal{T}$ is calculated as the sum of the distance threshold τ and the largest distance between a point in T_Q and its corresponding point in T'_Q. To ensure a fair comparison, the privacy budgets ϵ in NGram-filter and ATP-filter are normalized by the length of T_Q to maintain consistent interpretations with the other methods.

Metrics. We mainly assess the efficiency of GIST and baselines by the following metrics.

- **Running time.** It is the average response time for answering one FTM query.
- **Communication cost.** It is the total network communication between the query user and all data owners.
- **Retention rate.** It is the ratio of the size of the candidate trajectory database TC to the original database TD. The retention rate ranges between 0 and 1, with a smaller rate indicating a more effective filtering method.

Apart from the above three metrics, we also report the index size of our framework GIST in Sec. V-C.

Implementation. All the methods are implemented in C/C++ and compiled using GCC/G++ 9.4.0. We employ Obliv-C [22] for SMC operations and GMP library [28] for big integer computation. For all methods, we set the distance threshold $\tau = 50m$. In GIST, we set publishing rate $\rho = 60\%$, privacy parameters $\epsilon = 0.01, \delta = 2.5 \times 10^{-5}$, and partition parameter $\alpha = 0.5$. The query trajectory is randomly sampled from the dataset TD, and we reserve a subset of points in it. The portion of reserved points is controlled by the *sampling rate* parameter in Table III. We also vary other parameters, including trajectory scalability $|TD|$, privacy budget ϵ, partition parameter α, and the number of data owners. For each parameter setting, the average result of 100 queries is reported.



(a) Running time and communication cost on Geolife

(b) Running time and communication cost on Dazhong

(c) Running time and communication cost on Xi'an

Fig. 10: Running time and communication cost of FTM under different sampling rates of T_Q.

Environment. Experiments are carried out on 5 servers connected by LAN, each with 24 2.60GHz Intel(R) Xeon(R) Platinum 8361HC CPU processors and 128GB of memory.

B. Experiments on Real Datasets

To illustrate the efficiency of GIST in real-world applications, we conduct experiments on three real datasets, Geolife [23], Dazhong [25] and Xi'an [26].

Comparison Across Datasets. Fig. 10 shows the running time and communication cost of FTM in these real datasets. Four sampling rate are used: 5%, 10%, 20% and 40%. Across all three real datasets, GIST consistently outperforms the OblivC and the STSC-ext. In the Xi'an dataset [26] which contains 3.2 million trajectories, GIST is 19.8× to 420.8× faster than the runner-up PL-filter, while incurring 14.5× to 233.2× lower communication cost.

Vary Sampling Rate. The efficiency of GIST is primarily influenced by two factors: the number of trajectories requiring verification and the cost associated with securely verifying each trajectory. As the sampling rate increases, the effectiveness of filtering improves, leading to a reduction in the number of trajectories that need verification. Meanwhile, the cost of verifying each trajectory also increases. We observe that in Geolife dataset, the efficiency of GIST peaks when the sampling rate is 10%, while in Dazhong and Xi'an datasets, GIST achieves the best performance when the sampling rate is 40% and 20%, respectively. In contrast, the filtering rates of other methods change slightly as the sampling rate increases, since a higher sampling rate usually enlarges the safe threshold and undermines the filtering effectiveness. Accordingly, the running time of these methods exhibits an increasing trend.

TABLE IV: Construction time and memory size of grid index in GIST under different trajectory scalabilities.

Trajectory Scalability $\|TD\|$	1500k	3000k	4500k	6000k
Index Construction Time (s)	224.5	513.4	712.5	924.6
Index Size (MB)	154.8	321.7	448.1	585.2

C. Experiments on Scalability Test

We use Chengdu dataset [26] for scalability tests. Trajectory datasets TD of different sizes are generated by randomly sampling from the original dataset.

Vary Trajectory Scalability $|TD|$. Fig. 11 presents the results of scalability test under four levels of trajectory scalability $|TD|$. Generally, the running time and communication cost of GIST grow linearly with the data size, and maintain an advantage of 1 to 2 orders of magnitude over the best baseline at all sampling rates. Using sampling rate of 20% as an example, GIST is 147.5× to 186.8× more efficient than PL-filter, and takes 73.1× to 107.4× lower communication.

Vary Sampling Rate. As the sampling rate increases, the advantages of GIST become more evident. Taking trajectory scalability of 6 million as an example, GIST is 42.5× more efficient than STSC-ext when the sampling rate is 5%, and 526.8× more efficient when the sampling rate is 40%. This is because STSC-ext suffers from a prominent performance loss as the length of T_Q increases. In contrast, a longer query trajectory can be leveraged in GIST to enhance the filtering effectiveness, compensating for the performance loss in verification.

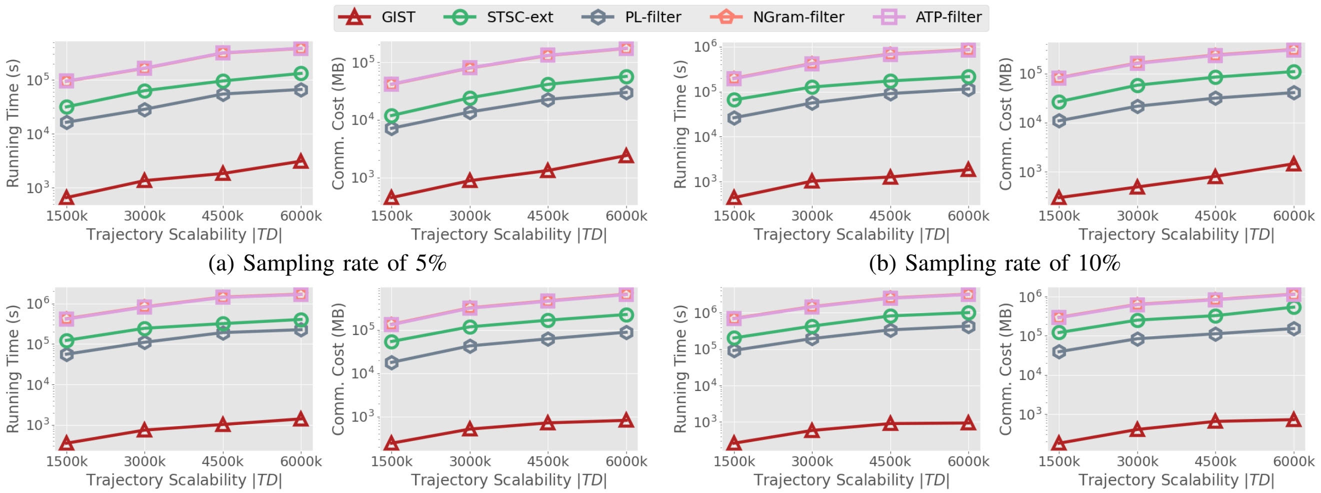
Index Construction. Table IV presents construction time and memory size of the grid index in GIST under different trajectory scalabilities. For the trajectory scalability of 6000k, the construction time is around 15min, which is feasible in the preprocessing stage. Besides, the grid index's size (585.2MB) is far more smaller than the size of the raw trajectory data (16.7G). The space cost is acceptable, considering the memory size of a modern server.

D. Experiments on Ablation Study

We conduct ablation studies from two aspects of our GIST: the *filtering* and *verification* phases. The following experiments are performed on Dazhong dataset [25], using 10% and 40% as the sampling rate for T_Q.

*1) **Ablation Study on Privacy Mechanism in Filtering**:* We compare the filtering effectiveness of GIST and baselines for location or trajectory privacy protection, including PL-filter [10], NGram-filter [11] and ATP-filter [12].

Vary Privacy Budget ϵ. Fig. 12 shows the retention rate of different filtering methods varying privacy budgets. It is demonstrated that for each method, a weaker privacy level indicated by larger ϵ, proves to be more effective in preserving the utility of the query trajectory, resulting in a lower retention



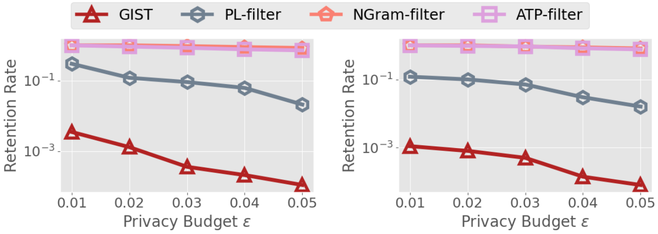
(a) Sampling rate of 5%

(b) Sampling rate of 10%

(c) Sampling rate of 20%

(d) Sampling rate of 40%

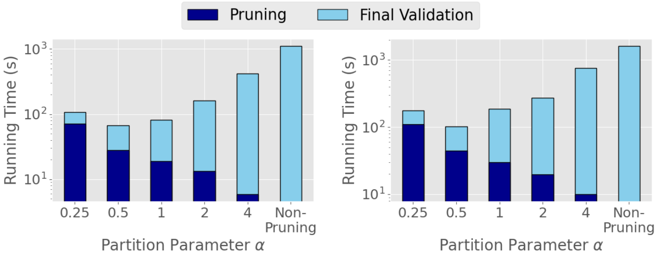
Fig. 11: Running time and communication cost of scalability tests on the Chengdu dataset.



(a) Sampling rate of 10%

(b) Sampling rate of 40%

Fig. 12: Retention rate of filtering in GIST and other trajectory publishing methods under different privacy budgets ϵ.



(a) Sampling rate of 10%

(b) Sampling rate of 40%

Fig. 13: Running time of pruning and final validation in SMC based verification under different partition parameters α.

rate. Notably, our filtering method demonstrates at least an 85× improvement in effectiveness under the same privacy budget. The results reveal that trajectories published using existing trajectory privacy mechanism cannot maintain high effectiveness when employed for filtering, illustrating the necessity of developing a novel privacy mechanism within our framework.

2) ***Ablation Study on Pruning in Verification***: The performance of SMC based verification is impacted by the partition parameter α. We compare the verification efficiency under different choices of α.

Vary Partition Parameter α. As shown in Fig. 13, pruning at all levels of α can reduce the running time of verification, illustrating the effectiveness of our pruning strategy. Besides, choosing a smaller α results in larger number of partitions, leading to increased pruning time but reduced final validation time. The figure indicates that in real-world data, pruning achieves the optimal performance when $\alpha = 0.5$ and bring an up to 16.2× improvement in running time.

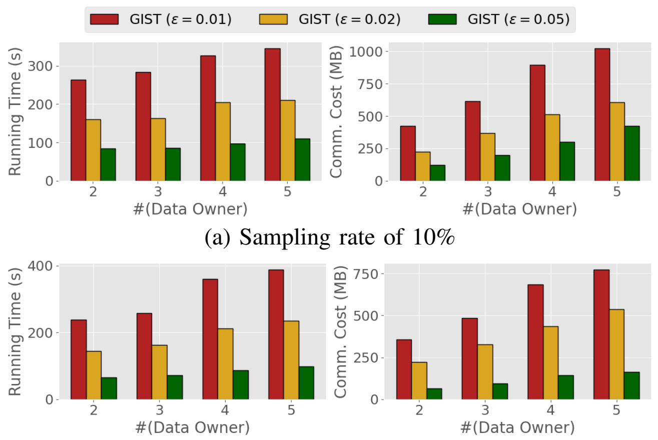
E. Experiments on Multiple Data Owners

Our GIST can be extended to a more general scenario where the trajectory database TD is distributed among multiple data owners. The extended method follows these steps: initially, the query user employs T_Q to generate G_Q and broadcasts it to all the data owners; then each data owner filters their local database using G_Q; finally, the query user performs verification with all the data owners in parallel. We conduct the experiment on Multi-Company dataset, a real-world trajectory dataset distributed among five data owners [18]. We report the performance of GIST with privacy budget ϵ of 0.01, 0.02 and 0.05 in the following experiments.

Vary #(Data Owner). The experiment results on Multi-Company dataset are shown in Fig. 14. Overall, the total communication cost of GIST for cross-platform data grow linearly with the number of data owners, while the running time remains relatively steady as the number of data owners increases. This is because the filtering and verification steps of GIST can be performed in parallel.

VI. Related Work

Trajectory Similarity Query. Various similarity measures have been proposed for trajectory data [16], [29]. Some measures consider spatial information only, such as DTW [30], ERP [31] and EDR [32]. Other measures consider both spatial and temporal information, such as STLCSS [33] and STED [15]. Based on different measures, existing works have designed efficient query processing solutions [34]–[36] and trajectory analytic systems [37]–[39]. However, they cannot be used for our FTM problem as they usually assume no



(a) Sampling rate of 10%

(b) Sampling rate of 40%

Fig. 14: Running time and communication cost of varying the number of data owners.

privacy protection for query users' or data owners' trajectories. Besides, directly combining them with the secure computation techniques is infeasible, since filtering and pruning strategy in these methods can lead to privacy leakage of query trajectory or trajectory database.

Trajectory Privacy Preservation. Privacy are crucial in trajectory analytics since trajectory data may disclose sensitive information like mobility patterns and personal profiles [1], [2]. *Geo-Indistinguishability* is widely used in protecting a user's location by injecting planar Laplacian noise, offering adaptive privacy preservation depending on the distance [10]. *Differential privacy* has also been applied in trajectory data publishing. Central differential privacy assumes all the trajectories are collected by a central server and publishes perturbed trajectories [40], [41] or synthetic trajectories [42], [43] with a statistical distribution similar to the original data. In contrast, local differential privacy does not rely on the central server and leverages exponential mechanism for privacy protection [11], [12]. Among these methods, we compare GIST with [10]–[12], since they are state-of-the-art solutions adaptable for filtering.

Data Federation Management. Data isolation has become an obstacle for cross-silo data analytics, since sharing raw data among data owners is usually prohibited due to privacy concerns [7], [44]. In response to these challenges, data federation has arisen as a promising paradigm, facilitating collaborative and secure query services for data owners interested in sharing their data. For example, SMCQL [17], Conclave [45] and Shrinkwrap [46] are data management systems over relational data federation. Hu-Fu [18] is a spatial data federation system. There are also studies on efficient processing of specific queries over a data federation, such as federated range aggregation [47], [48] and federated join [49], [50]. These works speed up the secure queries drastically by leveraging differential privacy to remove unnecessary dummy data/operations without sacrificing too much on the privacy. In comparison, our FTM problem differs from these studies in both data type and query type.

VII. Conclusion

In this paper, we study the problem of Federated Trajectory Matching (FTM) and introduce a framework called Geo-I accelerated SMC based method for federated Trajectory matching (GIST). We design a novel paradigm for publishing the query trajectory at a grid level and establish the bound for the grid size under specified privacy parameters. Besides, we devise a data partition scheme along with a reference trajectory based pruning strategy to further improve efficiency. Finally, experiments show that our method is significantly faster and takes up to 2 orders of magnitude lower communication cost than the state-of-the-arts.

Acknowledgment

This work is partially supported by National Science Foundation of China (NSFC) under Grant No. 2023YFF0725103, U21A20516, 62336003, and 62076017, Beijing Natural Science Foundation No. Z230001, CCF-Huawei Populus Grove Fund, Didi Collaborative Research Program NO2231122-00047, and the Beihang University Basic Research Funding No. YWF-22-L-531. Zimu Zhou's research is supported by Chow Sang Sang Group Research Fund No. 9229139.

References

[1] C.-Y. Chow and M. F. Mokbel, "Trajectory privacy in location-based services and data publication," in *SIGKDD Explorations Newsletter*, 2011.

[2] Y.-A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," in *Scientific reports*, 2013.

[3] Xinhua, "Digital maps help China track people flows amid epidemic," 2020. [Online]. Available: http://en.people.cn/n3/2020/0218/c90000-9658976.html

[4] BBC, "Safe cities- Using smart tech for public security," 2015. [Online]. Available: https://www.bbc.com/future/bespoke/specials/connected-world/government.html

[5] The General Data Protection Regulation (GDPR), 2016. [Online]. Available: https://eugdpr.org

[6] California Consumer Privacy Act (CCPA), 2018. [Online]. Available: https://www.caprivacy.org/

[7] Y. Zhang, Y. Li, Y. Wang, S. Wei, Y. Xu, and X. Shang, "Federated learning-outcome prediction with multi-layer privacy protection," in *Frontiers of Computer Science*, 2024.

[8] A. Liu, K. Zheng, L. Liz, G. Liu, L. Zhao, and X. Zhou, "Efficient secure similarity computation on encrypted trajectory data," in *ICDE*, 2015.

[9] Y. Lindell, "Secure multiparty computation," in *Communications of the ACM*, 2020.

[10] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *CCS*, 2013.

[11] T. Cunningham, G. Cormode, H. Ferhatosmanoglu, and D. Srivastava, "Real-world trajectory sharing with local differential privacy," in *PVLDB*, 2021.

[12] Y. Zhang, Q. Ye, R. Chen, H. Hu, and Q. Han, "Trajectory data collection with local differential privacy," in *PVLDB*, 2023.

[13] S. Wang, Z. Bao, J. S. Culpepper, and G. Cong, "A survey on trajectory data management, analytics, and learning," in *ACM Computing Surveys*, 2021.

[14] S. Šaltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez, "Indexing the positions of continuously moving objects," in *SIGMOD*, 2000.

[15] M. Nanni and D. Pedreschi, "Time-focused clustering of trajectories of moving objects," in *Journal of Intelligent Information Systems*, 2006.

[16] H. Su, S. Liu, B. Zheng, X. Zhou, and K. Zheng, "A survey of trajectory distance measures and performance evaluation," in *VLDB Journal*, 2020.

[17] J. Bater, G. Elliott, C. Eggen, S. Goel, A. N. Kho, and J. Rogers, "SMCQL: Secure query processing for private data networks," in *PVLDB*, 2017.

[18] Y. Tong, X. Pan, Y. Zeng, Y. Shi, C. Xue, Z. Zhou, X. Zhang, L. Chen, Y. Xu, K. Xu *et al.*, "Hu-Fu: Efficient and secure spatial queries over data federation," in *PVLDB*, 2022.

[19] F. Liu, Z. Zheng, Y. Shi, Y. Tong, and Y. Zhang, "A survey on federated learning: a perspective from multi-party computation," 2024.

[20] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, 2006.

[21] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," in *Foundations and Trends in Theoretical Computer Science*, 2014.

[22] Obliv-C, 2024. [Online]. Available: https://oblivc.org/

[23] Y. Zheng, X. Xie, W.-Y. Ma *et al.*, "GeoLife: A collaborative social networking service among user, location and trajectory," in *IEEE Data Engineering Bulletin*, 2010.

[24] Q. Liu, Y. Zeng, L. Chen, and X. Zheng, "Social-aware optimal electric vehicle charger deployment on road network," in *SIGSPATIAL*, 2019.

[25] SAIC Volkswagen, 2024. [Online]. Available: https://www.svw-volkswagen.com/

[26] Didi Chuxing, 2024. [Online]. Available: http://www.didichuxing.com/

[27] JinYinJian Technology, 2024. [Online]. Available: http://www.yinjian.com/

[28] GNU MP: The GNU Multiple Precision Arithmetic Library, 2024, http://gmplib.org/.

[29] D. Hu, L. Chen, H. Fang, Z. Fang, T. Li, and Y. Gao, "Spatio-temporal trajectory similarity measures: A comprehensive survey and quantitative study," in *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[30] B.-K. Yi, H. V. Jagadish, and C. Faloutsos, "Efficient retrieval of similar time sequences under time warping," in *ICDE*, 1998.

[31] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *VLDB*, 2004.

[32] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *SIGMOD*, 2005.

[33] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *ICDE*, 2002.

[34] D. Xie, F. Li, and J. M. Phillips, "Distributed trajectory similarity search," in *PVLDB*, 2017.

[35] S. Wang, Z. Bao, J. S. Culpepper, Z. Xie, Q. Liu, and X. Qin, "Torch: A search engine for trajectory data," in *SIGIR*, 2018.

[36] H. Yuan and G. Li, "Distributed in-memory trajectory similarity search and join on road network," in *ICDE*, 2019.

[37] Z. Shang, G. Li, and Z. Bao, "Dita: distributed in-memory trajectory analytics," in *SIGMOD*, 2018.

[38] Z. Fang, L. Chen, Y. Gao, L. Pan, and C. S. Jensen, "Dragoon: a hybrid and efficient big trajectory management system for offline and online analytics," in *VLDB Journal*, 2021.

[39] X. Ding, L. Chen, Y. Gao, C. S. Jensen, and H. Bao, "Ultraman: A unified platform for big trajectory data management and analytics," in *PVLDB*, 2018.

[40] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *CCS*, 2015.

[41] Y. Cao, Y. Xiao, L. Xiong, and L. Bai, "PriSTE: from location privacy to spatiotemporal event privacy," in *ICDE*, 2019.

[42] X. He, G. Cormode, A. Machanavajjhala, C. Procopiuc, and D. Srivastava, "Dpt: differentially private trajectory synthesis using hierarchical reference systems," in *PVLDB*, 2015.

[43] F. Jin, W. Hua, B. Ruan, and X. Zhou, "Frequency-based randomization for guaranteeing differential privacy in spatial trajectories," in *ICDE*, 2022.

[44] N. Sun, W. Wang, Y. Tong, and K. Liu, "Blockchain based federated learning for intrusion detection for internet of things," in *Frontiers of Computer Science*, 2024.

[45] N. Volgushev, M. Schwarzkopf, B. Getchell, M. Varia, A. Lapets, and A. Bestavros, "Conclave: secure multi-party computation on big data," in *EuroSys*, 2019.

[46] J. Bater, X. He, W. Ehrich, A. Machanavajjhala, and J. Rogers, "Shrinkwrap: efficient sql query processing in differentially private data federations," in *PVLDB*, 2018.

[47] Y. Shi, Y. Tong, Y. Zeng, Z. Zhou, B. Ding, and L. Chen, "Efficient approximate range aggregation over large-scale spatial data federation," in *IEEE Transactions on Knowledge and Data Engineering*, 2021.

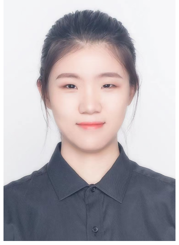
[48] M. Li, Y. Zeng, and L. Chen, "Efficient and accurate range counting on privacy-preserving spatial data federation," in *DASFAA*, 2023.

[49] Y. Wang and K. Yi, "Secure Yannakakis: Join-aggregate queries over private data," in *SIGMOD*, 2021.

[50] S. Li, Y. Zeng, Y. Wang, Y. Zhong, Z. Zhou, and Y. Tong, "An experimental study on federated equi-joins," in *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Yuxiang Wang is currently working toward the Master degree in the School of Computer Science and Engineering, Beihang University. His major research interests include federated data management, privacy-preserving data analytics, etc.

Yuxiang Zeng received the Ph.D. degree in computer science and engineering from the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology, in 2022. He is currently an associate professor in the School of Computer Science and Engineering, Beihang University. His research interests include spatio-temporal data management, federated data management, privacy-preserving data analytics, crowdsourcing, etc.



Shuyuan Li is currently working toward the Ph.D. degree in the School of Computer Science and Engineering, Beihang University. Her major research interests include federated data management, privacy-preserving data analytics, etc.

Yuanyuan Zhang received the PhD degree in computer science and technology from Beihang University in 2023. She is currently an engineer in North China Institute of Computing Technology. Her major research interests include federated learning, big spatiotemporal data mining and privacy preserving data analytics.

Zimu Zhou received the B.E. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2011, and the Ph.D. degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2015. He is currently an assistant professor at the School of Data Science, City University of Hong Kong. His research focuses on mobile and ubiquitous computing. He is a member of the IEEE.

Yongxin Tong received the Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology in 2014. He is currently a professor in the School of Computer Science and Engineering, Beihang University. His research interests include federated learning, federated data management, spatio-temporal data analytics, privacy-preserving data analytics, crowdsourcing, uncertain data management, etc. He is a member of the IEEE.