An Adaptive Real-time Trajectory Privacy Protection Algorithm

Yan Yan, Pengbin Yan*, Xinyu Kou, and Long Wang

Abstract—**With the rise of the mobile Internet, location-based services (LBS) have become increasingly popular. However, the location and trajectory leakage problem that can result from the use of LBS services makes protecting location privacy an increasing challenge. Existing approaches mainly achieve location protection by perturbing key location points or the whole trajectory, with little consideration of the correlation between locations, which may result in a significant decrease in the usability of the published trajectory. To address the aforementioned issues, this paper proposes a differential privacy trajectory-preserving method based on Mahalanobis distances. The proposed approach describes the correlation between neighboring location points by the amount of change in distance and direction. Then generates perturb locations by combining them into a covariance matrix using linear transformations of a non-uniform scaling matrix and an isometric transformations matrix. To better capture the directional correlation between adjacent locations within the trajectory, a specialized algorithm is designed to dynamically assign a regularization parameter based on the importance of the directional changes of adjacent locations. Experiments are carried out on different real trajectory datasets and compared with some existing methods to test the practicality of the proposed method. The experimental results and analysis indicate that the proposed method improves the utility of the published trajectory while ensuring its privacy.**

Index Terms—**Location-based services, Location privacy, Trajectory protection, Differential privacy, Mahalanobis distance**

I. Introduction

With the widespread popularity of mobile Internet, smart devices, and GPS, location-based services (LBS) have permeated every aspect of our daily lives [1], [2]. From navigation and route planning to point-of-interest queries and online car services, these conveniences rely on collecting and analyzing large amounts of location and trajectory data from users [3]. Location-based service providers (LBSPs) are dedicated to providing efficient and accurate services that allow users to enjoy convenient and personalized experiences based on their precise location. However, the very nature of such services requires LBSPs to collect and process sensitive geolocation information, which not only makes LBSPs a potential source of privacy threats but also potentially exposes users to a wide range of wireless communication threats [4]. User locations, if mishandled or accidentally leaked, can lead to several privacy and security concerns. For example, an attacker can obtain the real-time location of a user's trajectory by listening to or intercepting the communication data to perform tracking or other malicious and illegal activities. In addition, trajectory data often contains a significant amount of personal information. If accessed or publicized, this data could be exploited for social engineering attacks, advertising fraud, or even physical security threats. Furthermore, vulnerabilities in wireless communication [5], such as man-in-the-middle attacks and signal spoofing, can be exploited to interfere with or manipulate location data. The security vulnerabilities in smartphone applications [6] can also be leveraged by attackers to obtain trajectory information. Due to these privacy and security threats, LBSPs must adopt stringent data protection measures. Therefore, it has become imperative to find and implement appropriate privacy protection strategies to ensure the security of personal trajectory and location data.

To achieve the goal of trajectory privacy, most existing research employs the following approaches: the dummy trajectory method [7], [8], which does not publish the user's actual location in their trajectory, but instead generates a dummy location to obtain the corresponding service. The disadvantage of this approach is that it leads to a noticeable degradation in service quality. The generalization method [9], [10] improves privacy by decreasing the precision of location points within a trajectory, with the anonymity of the K trajectory being the primary representative technique. The more trajectories within the anonymized region, the stronger the privacy protection. The disadvantage of the generalization method is that when there is a large difference in the direction of the trajectory, it leads to an excessively large anonymization region, which consequently deteriorates the quality of the service provided. The suppression method [11], [12] enhances privacy by removing or concealing key locations within a trajectory. However, a significant drawback is that the level of privacy achieved largely depends on the number of sensitive locations that are suppressed. The differential privacy model [13] has gained considerable traction in the field of data publication, largely due to its rigorous mathematical definition. In recent years, a substantial body of research has emerged that explores the integration of differential privacy with trajectory privacy preservation, to develop a more comprehensive and efficient solution. Geo-indistinguishability [14] further extends the application of differential privacy models to the domain of location privacy protection. Subsequently, the extension of differential privacy in terms of metrics is called the differential metric privacy model [15]. Geo-indistinguishability not only safeguards

Manuscript received Nov. 4, 2024; revised April. 20, 2025. This work was supported by the National Nature Science Foundation of China (No. 62361036).

Yan Yan is a Professor of the School of Computer and Communication, Lanzhou University of Technology, Lanzhou,730050, China. (e-mail: yanyan@lut.edu.cn).

Pengbing Yan is a Postgraduate of the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, China. (e-mail: yanpb@lut.edu.cn).

Xinyu Kou is a Postgraduate of the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, China. (e-mail: 232085404070@lut.edu.cn).

Long Wang is a Postgraduate of the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, China. (e-mail: wanglong@lut.edu.cn).

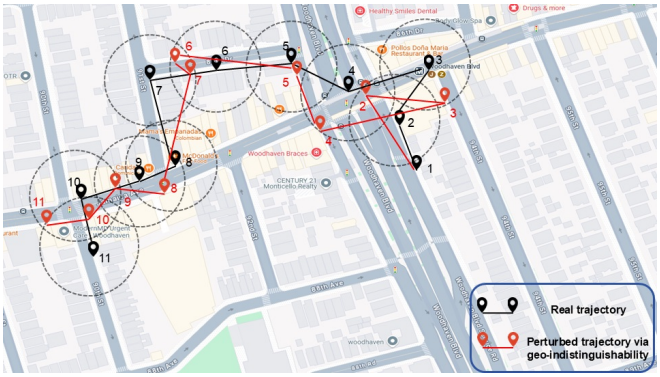


Fig. 1: An example of trajectory protection with geo-indistinguishability.

the privacy of individual locations but also enhances the overall privacy protection of the entire trajectories by adding controlled noise to the trajectory data.

Currently, differential location privacy perturbation methods for real-time trajectories have been less studied. Few studies take into account the correlation between neighboring locations within a trajectory. Take the road network environment shown in Fig. 1, the black line represents the user's real trajectory, while the red line is the perturbed trajectory generated through Geo-indistinguishability. Starting from the beginning of the real location and perturbing at the second and third locations, the generated perturbation trajectory is quite different from the direction of the real one. It can also be observed that at the end of the trajectory, the real and the perturbed locations are in different directions. This is because the Geo-indistinguishability perturbation mechanism with circular noise distribution is perceived the same for any direction. Therefore, it appears that, after applying a random perturbation, there may be points that seem disorganized and lack structure. This not only affects the effectiveness of real-time services but also disrupts the coherence of the trajectory, making it difficult to recognize the user's actual path. Therefore, when implementing perturbations to real-time trajectory locations, reasonable distances between neighboring points and continuous motion directions should be maintained to preserve the basic structure and utility of the trajectory.

To address the aforementioned issues of Geo-indistinguishability in real-time trajectory location perturbation, this paper proposes an adaptive real-time trajectory privacy-preserving algorithm. This introduces the correlation between neighboring locations in the trajectory into the design of the metric differential privacy perturbation algorithm, ensuring that the generated perturbed locations retain a greater degree of correlation with the preceding locations. Consequently, the usability of the published trajectory is enhanced.

II. Related work

Various solutions such as dummy trajectories, generalization methods, suppression methods, and differential privacy have been proposed to achieve privacy protection of trajectories. Wu et al. [16] proposed a virtual-based privacy-preserving algorithm that employs an adaptive virtual trajectory generation algorithm to generate uniformly distributed virtual trajectories, thereby fulfilling more privacy-preserving requirements with a reduced number of virtual trajectories. Shang et al. [17] proposed an algorithm that considers the case of trajectory similarity connection. This algorithm develops a pruning technique for the search space by defining a similarity concept and divides the algorithm into a two-phase partitioning algorithm to address the issues of trajectory duplicate detection, route planning, and traffic situation prediction. Hu et al. [18] proposed a trajectory protection algorithm for dividing the time intervals. The algorithm constructs an undirected trajectory graph by creating a privacy requirement matrix using trajectories that satisfy the specified constraints. It then constructs an undirected trajectory graph by identifying the relationship between the edges and the corresponding trajectories of vertices and creating K-anonymous sets to achieve privacy protection. Wang [19] proposed a novel approach to privacy protection, integrating social attributes into the protection process to address the growing threat of social privacy attacks. This technique responds to the limitation of current privacy protection methods for trajectories, which solely consider spatio-temporal attributes, neglecting the crucial role of social attributes. Tian et al. [20] proposed a personalized privacy trajectory protection technique, which extracts the distribution characteristics of trajectories by using Hilbert curves, proposes personalized algorithms for trajectories with different privacy preferences and achieves a trade-off between data privacy and utility. As differential privacy continues to evolve, an increasing number of researchers are investigating the integration of differential privacy with trajectory privacy protection, with the goal of developing a more comprehensive and efficient solution. Cheng et al. [21] proposed a differential privacy into the trajectory privacy protection algorithm and proposed the optimal personalized trajectory differential privacy algorithm. This algorithm first constructs a probabilistic mobility model of trajectories and semantic matching between different trajectories. It then proposes a privacy allocation scheme for trajectory identification through semantic similarity and finally perturbs the trajectory points. Sun et al. [22] improved the utility of the data by capturing individual motion patterns and combining them with differential privacy to generate trajectories that are closer to the real thing. Qiu et al. [23] proposed a privacy-preserving algorithm based on Geo-indistinguishability in order to address trajectory prediction attacks in continuous location queries. Dai et al. [24] proposed a differential privacy for grid anonymous trace privacy method. This method first extracts residency data from the trajectories and then assigns different privacy budgets according to different privacy levels to enhance the user's trajectory privacy.

After the concept of differential privacy, Geo-indistinguishability was proposed by Andres et al.[14], which is an application of differential privacy to two-dimensional locations. Their algorithm is an extension of differential privacy to the two-dimensional plane and an instance of metric differential privacy [15]. Geo-indistinguishability defines the obvious concept of introducing controlled noise to the user's exact location as a way to obtain a perturbed location, which is then sent to the LBS server in order to obtain the desired service. In the paper, Andres et al. use the planar Laplace algorithm to

achieve Geo-indistinguishability, where an attacker cannot distinguish between the true and perturbed locations within a circular area of radius r. Differential privacy has been extended to more research areas due to its flexibility [25], [26], [27], [28]. Xu [29] et al. proposed a differential privacy method using the regularized Mahalanobis metric to enhance privacy in text analysis scenarios. Zhao et al. [30] proposed the concept of directionality and combined the standard deviation ellipse to propose a new differential privacy algorithm, the Geo-ellipse-indistinguishability algorithm, which solved the loss of directional distribution in traditional Geo-indistinguishability and achieved higher directional distribution utility. Similar to the above algorithms, this paper also uses the differential privacy method based on the regularized Mahalanobis metric, but differently, we propose a dynamic trajectory privacy protection algorithm by combining it with the real-time trajectory privacy protection scenario and dynamically adjusting the parameters of the generated noise based on the criticality of the trajectory location points.

III. Preliminaries

Definition 1. (*ε-Differential Privacy* [13]). For sibling datasets D_1 and D_2 (D_1 and D_2 have the same attribute structure and differ by only one tuple) and any output $S \subseteq$ Range(*M*), given the privacy budget ε, if the probability that the mechanism *M* obtains the same output result on D_1 and D_2 satisfies the following inequality:

$$P_r\left[M\left(T_1\right) \in S\right] \leq e^{\varepsilon} \times P_r\left[M\left(T_2\right) \in S\right] \tag{1}$$

Then the mechanism is said to satisfy ε-differential privacy.

The differential privacy model uses data perturbation to ensure information security. It is grounded in rigorous mathematical theory and can quantitatively assess the level of privacy protection. In Eq. (1), the privacy budget parameter ε has a direct impact on the strength of privacy protection provided by the mechanism: a smaller value of this parameter means that the mechanism confers stronger privacy security; conversely, the level of privacy protection is relatively low. According to the differential privacy principle, even if an attacker obtains all information other than the target data entry, it is still impossible to accurately infer whether the entry is included in the original dataset or not, thus ensuring a high level of privacy security for user data.

Definition 2. (*Geo-indistinguishability* [14]). Given the privacy budget ε, if algorithm M obtains the same output z for any two location points x, x' and satisfies the following inequality, then the algorithm M satisfies Geo-indistinguishability:

$$M(\boldsymbol{x})(z) \leq e^{\varepsilon \mathrm{d}_2\left(\boldsymbol{x}, \boldsymbol{x}'\right)} M\left(\boldsymbol{x}'\right)(z) \tag{2}$$

where d_2 is the Euclidean distance.

Geo-indistinguishability extends the application of the differential privacy model from one-dimensional data to two-dimensional geographical locations and demonstrates significant advantages in location privacy protection. In the definition of Geo-indistinguishability, the privacy requirement is a constraint on the distance between geographical locations generated by two different location points x and x'. The Geo-indistinguishability model is a geographical extension of the differential privacy model, which only needs to replace the distance metric. This privacy protection method is called $d_{\mathcal{X}}$-metric differential privacy [15]. However, any distance metric must satisfy the following conditions:

$$d_{\mathcal{X}}\left(x, x'\right) \geq 0 \tag{3}$$

$$d_{\mathcal{X}}\left(x, x'\right) = d_{\mathcal{X}}\left(x', x\right) \tag{4}$$

$$d_{\mathcal{X}}\left(x, x'\right) \leq d_{\mathcal{X}}\left(x, z\right) + d_{\mathcal{X}}\left(z, x'\right) \tag{5}$$

Eq.(3) shows that the distance between any two locations cannot be negative, with the minimum distance being zero. Eq.(4) shows that distance does not matter to the order of two points. Eq.(5) shows that the sum of any two sides in a triangle is greater than or equal to the third side. It means that the distance between the two location points x and x' is less than or equal to the distance from location point x to the other location point z plus the distance from location point x' to location point z. Metric differential privacy is an effective extension of differential privacy, addressing the limitations of differential privacy in Hamming distance. The concept of metric differential privacy is as follows:

Definition 3. (*Metric differential privacy* [15]). Given the privacy budget ε, if algorithm M obtains the same output result z for any two locations x, x' and satisfies the following inequality, then the algorithm M satisfies metric differential privacy:

$$M(\boldsymbol{x})(z) \leq e^{\varepsilon \mathrm{d}_{\mathcal{X}}\left(\boldsymbol{x}, \boldsymbol{x}'\right)} M\left(\boldsymbol{x}'\right)(z) \tag{6}$$

Metric differential privacy can be applied in many fields. Since we only focus on location privacy in this article, locations are considered in the above definition.

Definition 4. (*Mahalanobis distance* [31]). Given two locations x_i = (lng_i, lat_i), x_{i-1} = (lng_{i-1}, lat_{i-1}), and a covariance matrix ϖ, the Mahalanobis distance is calculated as follows:

$$d_M\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \sqrt{\begin{pmatrix} \mathrm{lng}_i - \mathrm{lng}_j \\ \mathrm{lat}_i - \mathrm{lat}_j \end{pmatrix}^T \varpi^{-1} \begin{pmatrix} \mathrm{lng}_i - \mathrm{lng}_j \\ \mathrm{lat}_i - \mathrm{lat}_j \end{pmatrix}} \tag{7}$$

Where ϖ^{-1} is the inverse of the covariance matrix.

Definition 5. (*Regularized Mahalanobis distance* [31]). Given two locations x_i = (lng_i, lat_i), x_{i-1} = (lng_{i-1}, lat_{i-1}), and a covariance matrix ϖ, the regularized Mahalanobis distance is calculated as follows:

$$d_{RM}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \sqrt{\begin{pmatrix} \mathrm{lng}_i - \mathrm{lng}_j \\ \mathrm{lat}_i - \mathrm{lat}_j \end{pmatrix}^T \kappa^{-1} \begin{pmatrix} \mathrm{lng}_i - \mathrm{lng}_j \\ \mathrm{lat}_i - \mathrm{lat}_j \end{pmatrix}} \tag{8}$$

Where $\kappa = \{\lambda\varpi + (1-\lambda)I_m\}$, I_m is the identity matrix. if λ = 0, it is the Euclidean distance, if λ = 1, it is the Mahalanobis distance [31].

IV. Adaptive real-time trajectory privacy protection method

As can be seen from Fig. 1, the Geo-indistinguishability perturbation mechanism with a circular noise distribution has the same characteristics in every direction, which leads to the privacy protection mechanism not taking into account the user's movement direction and failing to capture important movement patterns or behavioral characteristics, which will

hurt traffic flow analysis, urban planning, behavior analysis, and other work. Furthermore, it can also hinder applications in accurately obtaining the user's movement direction, thus negatively impacting user experience.

The Geo-ellipse-indistinguishability proposed by Zhao et al. [30] utilizes the covariance matrix of global data to transform the circular noise distribution into an elliptical noise distribution. This transformation allows the algorithm to show differences in different directions and better preserve the correlation between location points. However, if the Geo-ellipse-indistinguishability is directly applied to a real-time dynamic scenario, the following problems will be caused. Firstly, in a real-time trajectory scenario, the user terminal cannot predict the user's next destination, nor can it access the user's global data. Secondly, analyzing the covariance matrix using historical data can result in a significant increase in time overhead, making it unsuitable for deployment on the user side. Furthermore, relying on early historical data may not produce advantageous outcomes.

Therefore, we propose an elliptic noise generation algorithm that can generate noise by constructing a covariance matrix according to the correlation of adjacent locations. In addition, we noticed that as the user's trajectory locations continuously change, the user's directional change significance also varies. Therefore, a dynamic adjustment algorithm is proposed for the elliptic noise perturbation algorithm, to retain more correlation in the locations with high significance, and reduce the retention of correlation in the locations with low significance.

A. Proposed Method

For location data, correlation can be described by distance magnitude and direction consistency between adjacent points. Therefore, the method proposed in this paper describes the change in distance between neighboring location points by constructing a non-uniform scaling matrix, and uses an isometric transformation matrix to represent the change in direction between neighboring location points. Finally, the covariance matrix is constructed by combining the linear transformations of the above two matrices so that the proposed method can perceive the correlation of the neighboring location points, which in turn improves the usability of the generated perturbation locations.

1) Non-uniform Scaling Matrix: Non-uniform scaling is a geometric transformation that applies different scaling factors in different dimensions. In mathematics and computer graphics, this transformation is often used to adjust the shape of an object. For example, in two dimensions, an object may be stretched horizontally while remaining unchanged vertically; in three dimensions, different scaling factors can be applied to the length, width, and height, respectively. In two dimensions, a non-uniform scaling transformation can be implemented by a special linear transformation matrix that expresses a change in the dimensions of a point or vector while keeping its shape and scale constant. This transformation takes the locations (X_i, Y_i) as an example, which can be expressed as follows:

$$\begin{pmatrix} \mathrm{X}_i' \\ \mathrm{Y}_i' \end{pmatrix} = \begin{pmatrix} S_{\mathrm{X}} & 0 \\ 0 & S_{\mathrm{Y}} \end{pmatrix} \begin{pmatrix} \mathrm{X}_i \\ \mathrm{Y}_i \end{pmatrix} \tag{9}$$

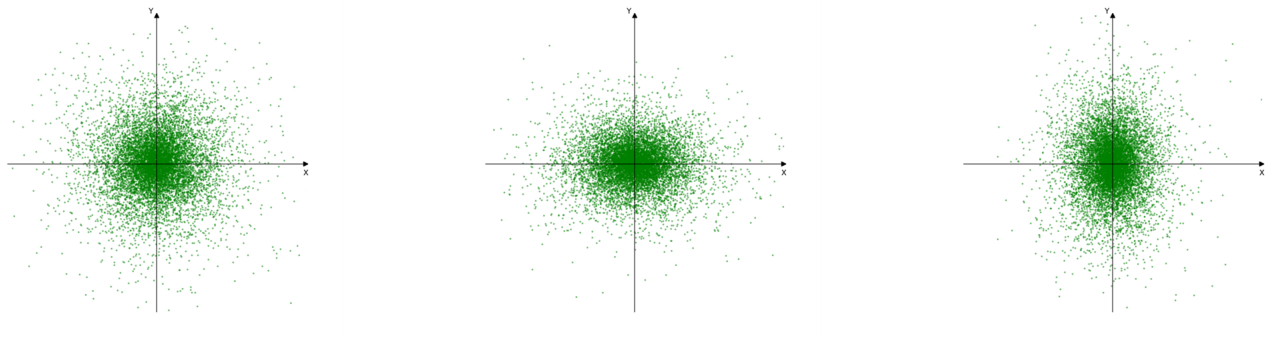
where $\begin{pmatrix} X_i \\ Y_i \end{pmatrix}$ represents the coordinates of the original location point, S_X and S_Y are the scale factors along the longitude and latitude directions respectively, $\begin{pmatrix} X_i' \\ Y_i' \end{pmatrix}$ represents the coordinates of the transformed location point. When S_X and S_Y are both equal to 1, the distribution of the generated noise is circular, as shown in Fig. 2(a). When S_X = 1 and S_Y = 0.5, the object remains unchanged along the longitude axis, the size decreases along the latitude axis, and the distribution of the generated noise is elliptical, as shown in Fig. 2(b). While S_X = 0.5 and S_Y = 1, the object remains unchanged along the latitude axis and the size decreases along the longitude axis, as shown in Fig. 2(c). Therefore, the shape of the noise distribution at the perturbation location can be adjusted from circular to elliptical with the help of a non-uniform scaling matrix, while S_X and S_Y can determine the semi-major and semi-minor axes of the elliptical noise distribution, which determines the eccentricity and thus the shape of the ellipse. The non-uniform scaling matrix S is:

$$S = \begin{pmatrix} S_X & 0 \\ 0 & S_Y \end{pmatrix} \tag{10}$$

After knowing how the non-uniform scaling matrix is obtained, the semi-major and semi-minor axes of the elliptical noise distribution need to be determined. In the elliptical noise distribution, the semi-major and semi-minor axes determine the eccentricity of the elliptical noise distribution, while the values of S_X and S_Y in the non-uniform scaling matrix determine the semi-major and semi-minor axes of the elliptical noise distribution. In order to perceive the latitude and longitude changes of neighboring location points, this paper proposes a way to set the values of S_X and S_Y according to the changes in the horizontal and vertical coordinates of the location points, which treats the coordinate axes with large variations as the semi-major axes in the elliptical noise distribution to produce a more dispersed noise, while the coordinate axes with small variations are treated as the semi-minor axes to produce denser noise. The resulting noise then provides more privacy on axes with large changes and boosts more correlation utility on axes with small changes. Specifically, the coordinates (X_i, Y_i,) and (X_{i-1}, Y_{i-1},) of the location points are obtained by projecting the current location $x_{i-1} = (lng_{i-1}, lat_{i-1})$ and the previous location $x_i = (lng_i, lat_i)$ onto the Cartesian coordinate system. The coordinate difference between the current location coordinates and the previous location point is computed and denoted as S_X and S_Y, respectively. Since S_X and S_Y are the coordinate differences between two locations in the Cartesian coordinate system, the values of S_X and S_Y may be very large. The method in this paper is to change the circular noise produced by the unit matrix into elliptical noise distribution, so the value of the non-uniform scaling matrix cannot be set too large. It is necessary to adjust S_X and S_Y again according to the following formula:

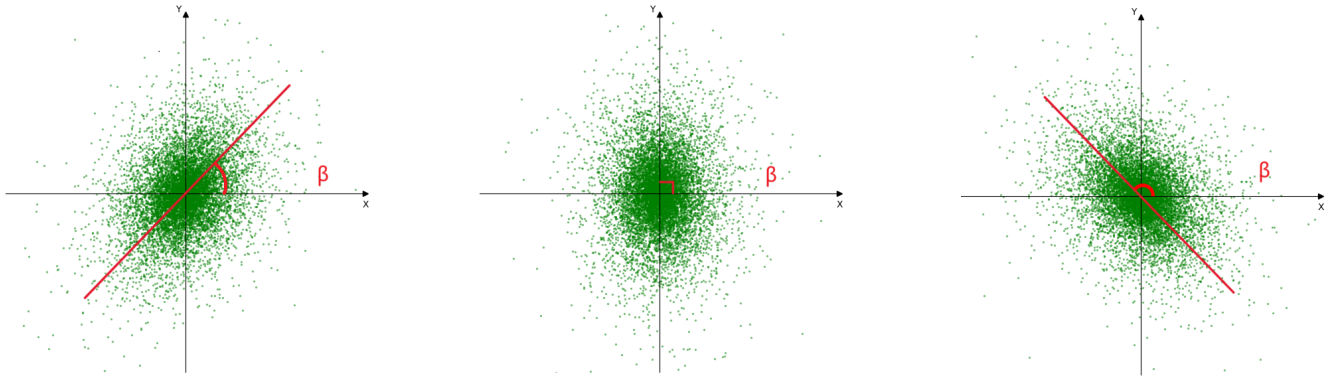
$$\begin{cases} S_X = S_Y = 1 & \text{if } S_X = S_Y \\ S_X = \max\left\{\frac{S_X}{S_Y}, 0.2\right\}, S_Y = 1 & \text{if } S_X < S_Y \\ S_X = 1, S_Y = \max\left\{\frac{S_Y}{S_X}, 0.2\right\} & \text{if } S_X > S_Y \end{cases} \tag{11}$$

After obtaining S_X and S_Y by calculation, if the value



(a) S_x = 1, S_y = 1 (b) S_x = 1, S_y = 0.5 (c) S_x = 0.5, S_y = 1

Fig. 2: Effect of non-uniform scaling.



(a) β = 45° (b) β = 90° (c) β = 135°

Fig. 3: Effect of isometric transformations.

of S_Y is greater than S_X, indicating that the change of Y-axis coordinates is slightly larger, then set S_Y = 1 and S_X = S_X/S_Y, to prevent the value of S_X/S_Y from being too small and resulting in the noise points showing a linear distribution. Thus set S_X to be $max\{S_X/S_Y, 0.2\}$; if the value of S_X is greater than S_Y, then set S_X = 1 and $max\{S_Y/S_X, 0.2\}$.

2) Isometric Transformation Matrix: When using the non-uniform scaling matrix set up in the previous section, the proposed algorithm produces more reasonable noise only when the user is moving in a straight line. In the real world, the user does not move in a straight line all the time, and the algorithm does not work well when the user's trajectory is skewed. Therefore, this subsection introduces isometric transformations to rotate the noise so that the user trajectory works properly when it is tilted.

Isometric transformations are a special class of geometric transformations that keep the distance between points constant. In two or three dimensions, this usually consists of the following basic types of operations: translation and rotation. Both translations and rotations are performed without affecting the distances between distributions. The shape of the noise distribution at the perturbed location can be adjusted using a non-uniform scaling matrix, while the direction of the noise distribution at the perturbed location can be further rotationally transformed with the help of isometric transformations. Therefore, in order to enable a better perception of the change in direction between neighboring locations, the angle β between the line connecting the neighboring locations and the positive x-axis needs to be calculated. After obtaining the rotation angle β parameter, one can use the isometric transformation matrix R as in Eq. (12) to rotate the major axis of the ellipse to align with the orientation of the neighboring locations. However, since this is done by rotating the major axis of the ellipse noise distribution, it results in a weakened perception of the differences in latitude and longitude.

$$R = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \tag{12}$$

In order to visualize the effect of rotation, S_X = 1 and S_Y = 0.5 in the fixed non-uniform scaling matrix are fixed, and the principle of rotational transformations under isometric transformations is demonstrated by choosing different angles, e.g., 45°, 90°, and 135°. In Fig. 3, it can be observed that the shape and size of the ellipse remain the same during the rotation, only the direction changes.

3) Perturbation Location Generation Algorithm: After obtaining the non-uniform scaling matrix S and the isometric transformations matrix R, the covariance matrix can be constructed by combining the following linear transformations:

$$\varpi = RSR^T \tag{13}$$

where R^T is the transpose matrix of the isometric transformations matrix.

Algorithm 1 describes the pseudo-code of the elliptical noise distribution generation algorithm proposed in this paper. The inputs to the algorithm are, in order, the coordinates of the previous location (X_{i-1},Y_{i-1}) after projection through the Cartesian coordinate system, the coordinates of the current location (X_i, Y_i), the privacy budget ε, and the regularization parameter λ. The output of the algorithm is the perturbed Cartesian coordinates (X_i, Y_i). Steps 1 and 2 samples from a multivariate normal distribution with a mean of (0, 0) and a covariance matrix of the 2x2 identity matrix to obtain the vector v. Normalize v to obtain a vector w. Steps 3 to 5 calculate β needed for the isometric transformation matrix R and S_X and S_Y needed for the non-uniform scaling matrix S. Then, combine R and S to form the covariance matrix. Step 6 samples from the gamma distribution with shape 2 and parameter 1/ε to get r. Steps 7 to 9 apply the noise value obtained from the linear variation to be added to the current location. The regularization parameter in step 7 will be described in detail in the next section.

Algorithm 1 Elliptical noise distribution generation algorithm.

Input: previous moment Cartesian coordinates (X_i-1, Y_i-1), current moment Cartesian coordinates (X_i, Y_i), privacy budget ε, regularization parameter λ

Output: current moment after perturbation Cartesian ($\overline{X_i}$,$\overline{Y_i}$)

1: Sample v from a multivariate normal distribution with a mean of (0, 0) and a 2x2 identity matrix as the covariance matrix.
2: Normalize v obtain w.
3: Calculate the angle to the positive x-axis from (X_{i-1}, Y_{i-1}) and (X_i, Y_i), to get the rotation angle , and then get R.
4: S_X and S_Y are computed from (X_{i-1}, Y_{i-1}) and (X_i, Y_i), and then S.
5: Obtain ϖ using R,S matrix.
6: Sample r from a gamma distribution with shape parameter 2 and scale parameter 1/ε.
7: $NX, NY = r\{\lambda\varpi + (1-\lambda)I_2\}^{1/2} w$
8: ($\overline{X_i}$,$\overline{Y_i}$) = (X_i, Y_i) + (NX, NY).
9: return ($\overline{X_i}$,$\overline{Y_i}$)

B. Dynamic Adaptive Parameter Tuning

In Section 3, it was mentioned that the parameter λ of the regularized Mahalanobis distance adjusts the semi-major and semi-minor axes of the elliptical noise distribution produced by Algorithm 1. Consequently, it also affects the eccentricity of the resulting elliptical noise distribution. As the value of λ gets closer to 1, the closer the elliptical noise distribution is to the correlation between neighboring locations, and as it gets closer to 0, the less correlation is retained to neighboring locations. Fig. 4 depicts the schematic diagram for λ of 0, 0.5, and 1, respectively:

As illustrated in Fig. 4, when λ = 0, the noise distribution appears circular, and the algorithm has no perception of the correlation between adjacent locations. When λ = 0.5, the noise distribution gradually elongates into an ellipse with lower eccentricity, and the algorithm's perception of correlation between adjacent locations increases. When λ = 1, the elliptical noise distribution becomes more pronounced, and the algorithm's perception of correlation between adjacent locations reaches its maximum. However, in trajectory scenarios, not all trajectory points require a high level of correlation retention. Therefore, we can adjust the λ value based on the need for correlation retention, increasing the λ value at points where high correlation retention is required, and decreasing it where high correlation is unnecessary.

Combining the concept of key points in a trajectory proposed in a related study [23], we propose directional change significance as a way to determine those location points that need to retain higher relevance. As shown in Fig. 5, the directional change significance of location point 3 is equal to the ratio of the magnitude of the α_3 pinch angle to the maximum angle. The larger the angle, the greater the directional change significance, e.g., α_7; the smaller the angle, the smaller the directional change significance. The higher the directional change significance of a location point, the better it reflects the overall architecture of the user's trajectory.

Therefore, we propose a dynamic λ parameter tuning scheme that makes the directional change significance grow in parallel with the λ parameter, that is, the λ parameter is proportional to the directional change significance. Since $\lambda \in [0, 1]$, the specific formula is as follows:

$$\lambda_i = \frac{\alpha_i}{\pi} \tag{14}$$

For example, when the angle α = $\frac{\pi}{4}$, λ = $\frac{1}{4}$, at which time the value of λ is small, which means that preserving the correlation between neighboring location points is on the low side; while when the angle α = $\frac{3\pi}{4}$, λ = $\frac{3}{4}$, at which time the value of λ is large, which also means that preserving the correlation between neighboring location points is on the high side.

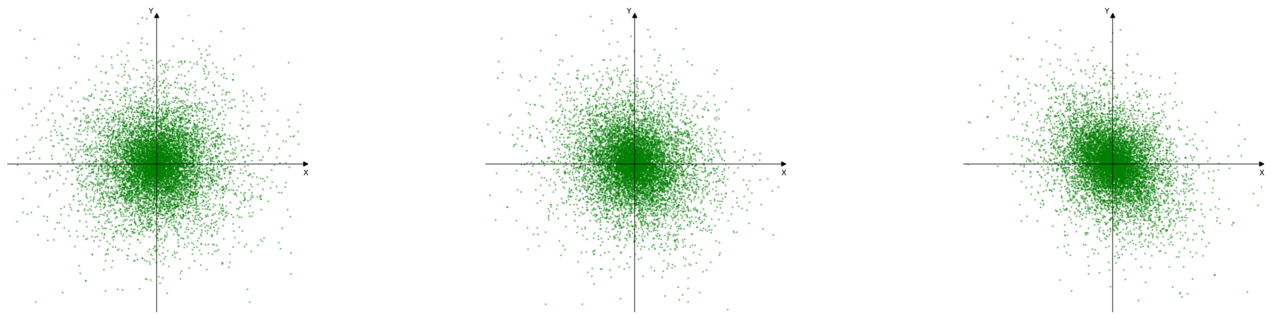
Algorithm 2 Adaptive real-time trajectory privacy protection algorithm (ARTPP).

Input: Set of real trajectory Cartesian coordinates (X_1, Y_1), (X_2, Y_2),....., (X_n, Y_n) ,Privacy budget ε

Output: Set of Cartesian coordinates for perturbed trajectories ($\overline{X_1}$,$\overline{Y_1}$), ($\overline{X_2}$,$\overline{Y_2}$),....., ($\overline{X_n}$,$\overline{Y_n}$)

1: **for** i $\in \{3, ..., n\}$ **do**
2: Calculate from (X_{i-2}, Y_{i-2}), (X_{i-1}, Y_{i-1}), (X_i, Y_i).
3: Calculate λ_i using $\frac{\alpha_i}{\pi}$.
4: Sample $(\overline{X_i}, \overline{Y_i})$ using Algorithm 1.
5: **end for**
6: return ($\overline{X_1}$,$\overline{Y_1}$), ($\overline{X_2}$,$\overline{Y_2}$),....., ($\overline{X_n}$,$\overline{Y_n}$)

Algorithm 2 describes the pseudo-code of the proposed Adaptive real-time trajectory privacy protection algorithm (ARTPP). The inputs of Algorithm 2 are sequentially the set of user trajectory locations projected through the Cartesian coordinate system and the privacy budget ε. The output of Algorithm 2 is the set of user trajectory locations in Cartesian coordinates after perturbation. Since the algorithm needs to calculate the pinch angle α, it must involve three locations. Therefore, for the first location point, no perturbation is performed, and the second location point is processed by adding noise using Algorithm 1. From the third location,



(a) λ = 0 (b) λ = 0.5 (c) λ = 1

Fig. 4: Effect of adjusting the parameter λ.

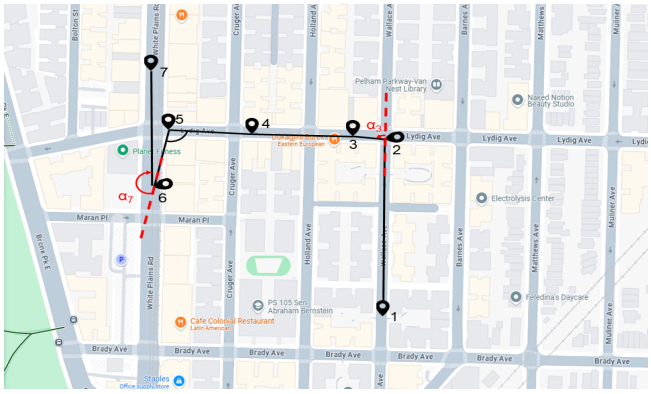


Fig. 5: Schematic of directional change significance.

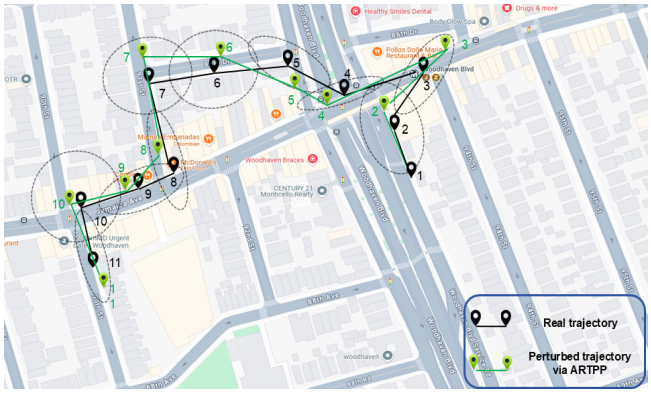


Fig. 6: Real trajectory protection effect using ARTPP algorithm.

the angle λ_i is calculated. Once this angle is obtained, Algorithm 1 is input to generate the perturbed location until the whole trajectory has been processed. Fig. 6 depicts the real trajectory protection effect using the proposed ARTPP algorithm.

C. Proof of Privacy

Theorem 1. For any given $\varepsilon > 0$, $\lambda \in [0, 1]$, Algorithm 1 satisfies $\varepsilon d_{RM} - privacy$.

Proof: To prove theorem 1, we have to prove the following inequality:

$$M(\boldsymbol{x})(z) \leq e^{\varepsilon \mathrm{d}_{RM}\left(\boldsymbol{x}, \boldsymbol{x}'\right)} M\left(\boldsymbol{x}'\right)(z)$$

By inputting the probability density function, the following results are obtained:

$$\frac{\Pr[M(x)(z)]}{\Pr\left[M\left(x'\right)(z)\right]} = \frac{e^{-\varepsilon\sqrt{(x-z)^T\{\lambda\varpi+(1-\lambda)I_m\}^{-1}(x-z)}}}{e^{-\varepsilon\sqrt{(x'-z)^T\{\lambda\varpi+(1-\lambda)I_m\}^{-1}(x'-z)}}} \leq e^{\varepsilon\sqrt{(x-x')^T\{\lambda\varpi+(1-\lambda)I_m\}^{-1}(x-x')}}$$

The above inequality can be simplified as:

$$-\varepsilon\sqrt{(x-z)^T\left\{\lambda\varpi+(1-\lambda)I_m\right\}^{-1}(x-z)}$$

$$+\varepsilon\sqrt{(x'-z)^T\left\{\lambda\varpi+(1-\lambda)I_m\right\}^{-1}(x'-z)}$$

$$\leq \varepsilon\sqrt{(x-x')^T\left\{\lambda\varpi+(1-\lambda)I_m\right\}^{-1}(x-x')}$$

Since ϖ in Algorithm 1 is formed by a combination of linear transformations and satisfies positive definite, eigendecomposition can be used to obtain:

$$\varpi = Q\Lambda Q^T$$

There is $Q^{-1} = Q^T$, and Λ is a diagonal matrix with all elements greater than 0, so that:

$$\{\lambda\varpi + (1-\lambda)I_m\} = \{\lambda Q\Lambda Q^T + (1-\lambda)I_m\}$$

Since $Q^TQ = I$, we can simplify as: $Q\{\lambda\varpi + (1-\lambda)I_m\}Q^T$. Now, we seek $\{\lambda\varpi + (1-\lambda)I_m\}^{-1}$.

Since $\{\lambda\varpi + (1-\lambda)I_m\}$ is a diagonalizable matrix, its inverse is the reciprocal of each element on the diagonal, let

$$\nu^{-1} = \lambda\Lambda + (1-\lambda)I_m$$

Therefore, the final inverse matrix is:

$$Q\{\lambda\varpi + (1-\lambda)I_m\}^{-1}Q^T = Q\nu Q^T$$

The matrix $Q\nu Q^T$ can be replaced with a new covariance matrix M, and the equivalent expression can be rewritten as:

$$-\varepsilon\sqrt{(x-z)^TM(x-z)} + \varepsilon\sqrt{(x'-z)^TM(x'-z)} \leq \varepsilon\sqrt{(x-x')^TM(x-x')}$$

Since the Mahalanobis distance measure satisfies the triangle inequality, we can thus prove that Algorithm 1 satisfies εd_{RM}-privacy.

V. Experiments and Analysis

In order to fully evaluate and analyze the proposed algorithms, it was compared with some related algorithms in recent years in terms of usability and directionality of the published trajectory. These algorithms include Geo-indistinguishability(Geo-ind) and Geo-ellipse-indistinguishability(Geo-ellipse-ind). It is important to note that when λ = 0, our proposed algorithm has the same principle as Geo-indistinguishability.

The experimental environment is a 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz, 32 GB of RAM, Windows 11 operating system. The algorithm is implemented using Python 3.11.

A. Experimental Dataset

The T-drive dataset and the Geolife dataset are selected to carry out the experiments and comparisons. The T-drive dataset contains the GPS trajectories of 10,357 cabs in Beijing from February 2 to February 8, 2008. During the experiments, it was firstly filtered by longitude range [116.35, 116.45] and latitude range [39.85, 40.0], and then, the trajectory data of 1,200 cabs during February 8, 2008, was selected as our experimental data. The Geolife dataset contains the GPS trajectory data of 182 users in more than three years. During the experiments, each user's trajectory in the dataset with the period of 8:00 a.m. to 10:00 a.m. was selected to be the experimental data.

B. Evaluation Indicators

Trajectory privacy protection methods based on perturbation strategies usually use the perturbed locations instead of users' real coordinates so as to realize location-based services without exposing users' location privacy. In this framework, the spatial distance between the perturbed location and the actual location becomes an intuitive indicator of the loss of location service quality. To quantify the change in location service quality caused by different privacy protection strategies, this paper adopts the average of the spatial distance between the perturbed location and the real location of all the trajectory location points, also known as the distance error, as an evaluation index, which is defined as:

$$\text{Distance}_{\text{error}} = \frac{1}{n} \times \sum_{i=1}^{n} d\left(x_i, x_i^*\right) \tag{15}$$

where n is the length of the trajectory.

In addition, to evaluate the directionality error between trajectories, we use the directionality error between neighboring points and the directional consistency index defined below as evaluation metrics. The directionality error between neighboring points is defined as follows:

Let $Tr = \{(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)\}$ be the real trajectory dataset, the perturbation data set is $Tr^{'} = \left\{(X_1^{'}, Y_1^{'}), (X_2^{'}, Y_2^{'}), ..., (X_n^{'}, Y_n^{'})\right\}$. Then, the polar angle between neighboring points of dataset Tr can be described as:

$$Tr\_\alpha_i = \arctan\left(\frac{Y_i - Y_{i-1}}{X_i - X_{i-1}}\right) \tag{16}$$

The polar angle between neighboring points of dataset Tr' can be expressed as:

$$Tr'\_\theta_i = \arctan\left(\frac{Y_i' - Y_{i-1}'}{X_i' - X_{i-1}'}\right) \tag{17}$$

The directionality error between neighboring points of the real trajectory dataset and the perturbed dataset is formulated as follows:

$$\text{Directionality}_{\text{error}} = \frac{\sum |Tr\_\alpha_i - Tr'\_\theta_i|}{n-1} \tag{18}$$

The directional consistency index (DCI) is used to measure the consistency of two trajectories in terms of direction. By calculating the initial heading between neighboring locations of the real trajectory plus the preset threshold angle to get the angular error tolerance range, and then calculating whether the direction of the locations of the perturbed trajectory is in the angular error tolerance range. The DCI indicates that among all the trajectory points, the number of directional discrepancies in the tolerance range is in the proportion of the whole. The index is expressed as a percentage, with higher values indicating that the two trajectories are more consistent in direction. The initial heading indicates the direction of the shortest path along the Earth's surface from the first location to the second location.

Let the real trajectory data and the perturbed trajectory, each with N corresponding trajectory points, have initial heading angles of Θ_i^{real} and Θ_i^{pertu}, respectively, where i = 1, 2, ..., N. Setting a preset threshold angle $\Delta\Theta$, the direction consistency index is calculated as follows:

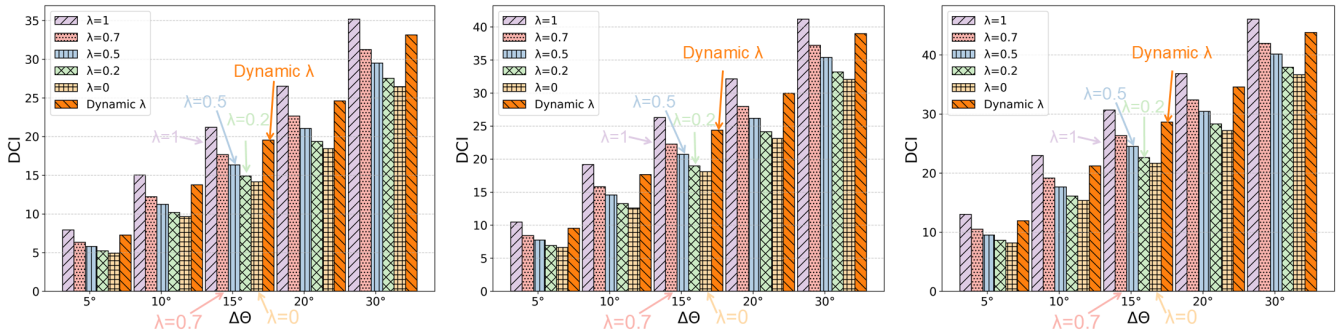
$$\text{DCI} = \frac{N_{\text{condition}}}{N} \times 100\% \tag{19}$$

where $N_{\text{condition}}$ represents the number of points that satisfy the condition:

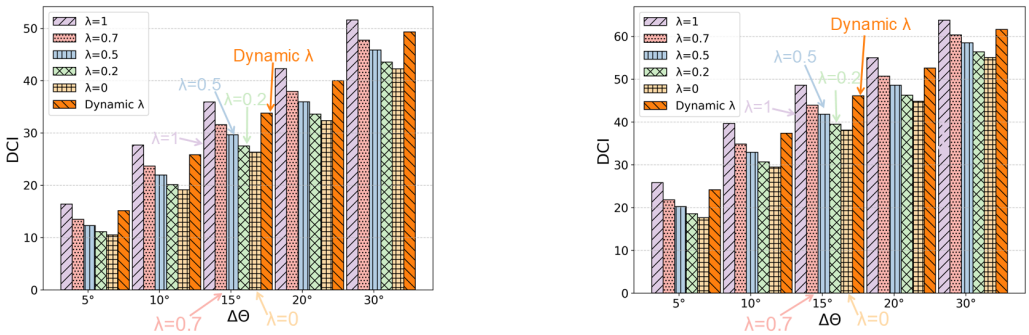
$$|\Delta\vartheta_i| = |\vartheta_i^{\text{real}} - \vartheta_i^{\text{pertu}}| \leq \Delta\Theta \tag{20}$$

C. Results of the Experiments

1) Experimental Results of the Proposed Method: Select privacy budget ε in [0.003,0.005,0.007,0.01,0.02], and set the preset angle thresholds at [5°, 10°, 15°, 20°, 30°] for the test to observe the changes in the evaluation index under different parameter settings. Fig. 7 and Fig. 8 depict the comparison of DCI with different λ on the experimental datasets. No matter how much the preset angle threshold is increased, the parameter λ = 0 has the lowest DCI value and λ = 1 has the highest DCI value. When λ is dynamic changed, the DCI value is between λ = 1 and λ = 0.7. The above phenomenon proves that the parameter λ can adjust the eccentricity of the elliptical noise distribution, thus providing a better correlation utility at larger values. With the constant increase of the preset angle threshold, it is obvious that the DCI value of all λ values is also increasing. Because the larger the angle, the more disturbance location points will be included. As ε keeps increasing, the DCI value gets progressively higher, this is because ε is a parameter used to adjust the size of the noise, the larger ε adds the smaller noise. So, the DCI value gets progressively higher. These experimental results show that the parameter configurations

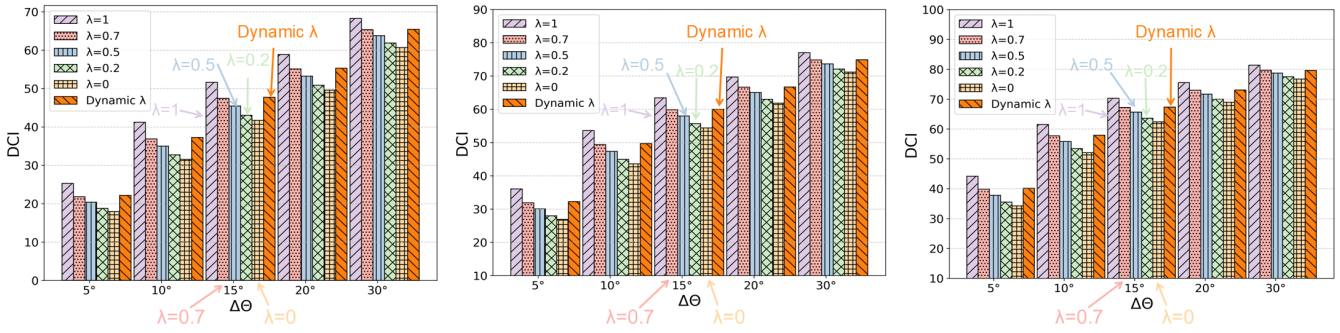


(a) ε = 0.003 (b) ε = 0.005 (c) ε = 0.007

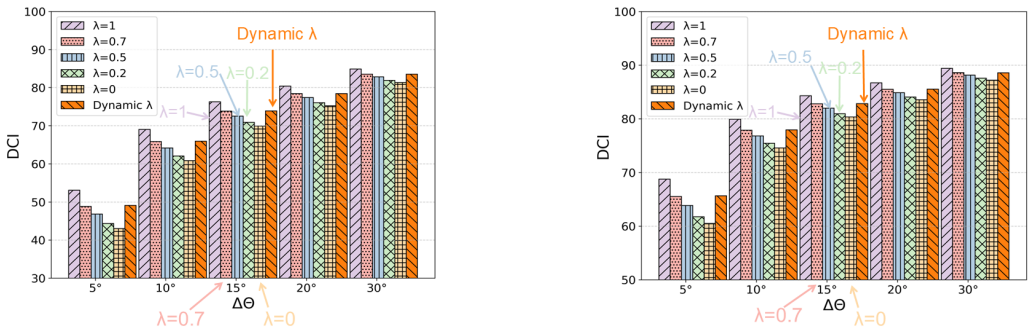


(d) ε = 0.01 (e) ε = 0.02

Fig. 7: Comparison of DCI with different λ on Geolife dataset.



(a) ε = 0.003 (b) ε = 0.005 (c) ε = 0.007

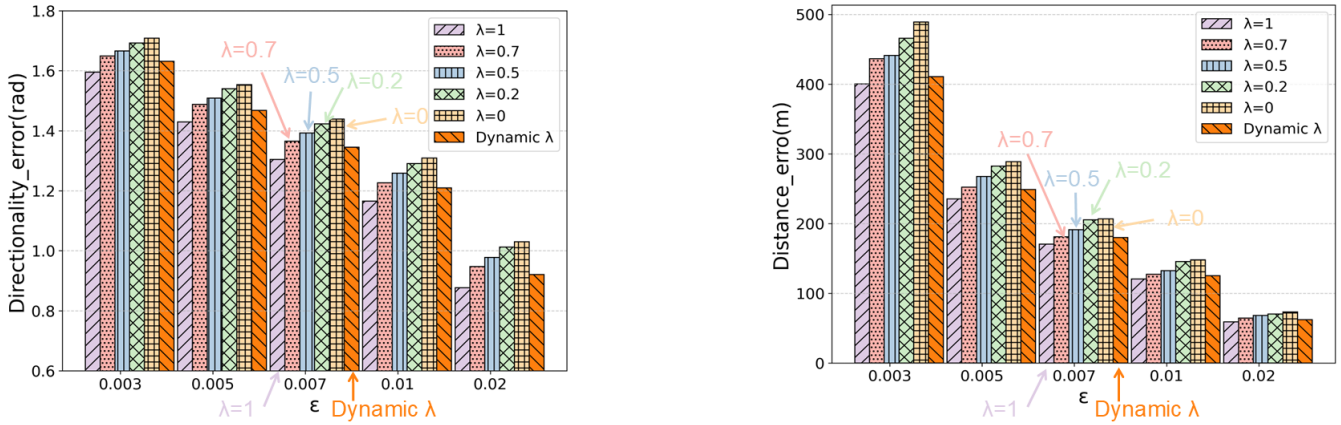


(d) ε = 0.01 (e) ε = 0.02

Fig. 8: Comparison of DCI with different λ on T-drive dataset.

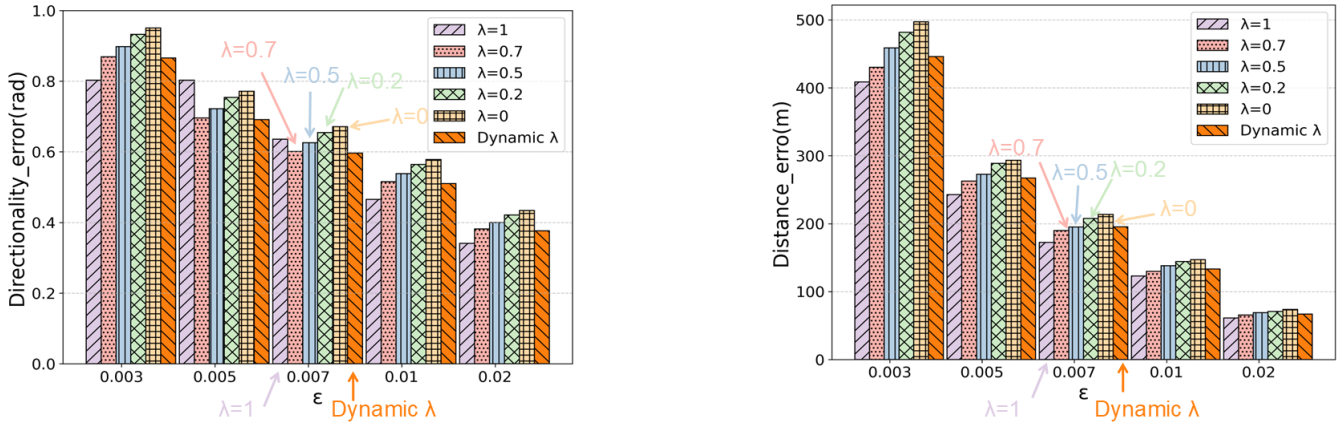
have similar effects on the balance of privacy protection and data utility in different location datasets.

Then, let's focus on the variation of the proposed method in terms of distance error and directional error between neighboring points. Related experimental results are portrayed in Fig. 9 and Fig. 10. As depicted in Fig. 9 (a) and Fig. 10 (a), regardless of ε, the parameter $\lambda = 0$ has the largest directionality error and $\lambda = 1$ has the smallest directionality error, whereas when it is an adaptive dynamic variation, the directionality error is presented between $\lambda = 1$ and $\lambda = 0.7$. The above phenomenons are also due to the fact that the parameter adjusts the eccentricity of the elliptical



(a) Directionality$_{error}$ (b) Distance$_{error}$

Fig. 9: Comparison of errors with different λ on Geolife dataset.



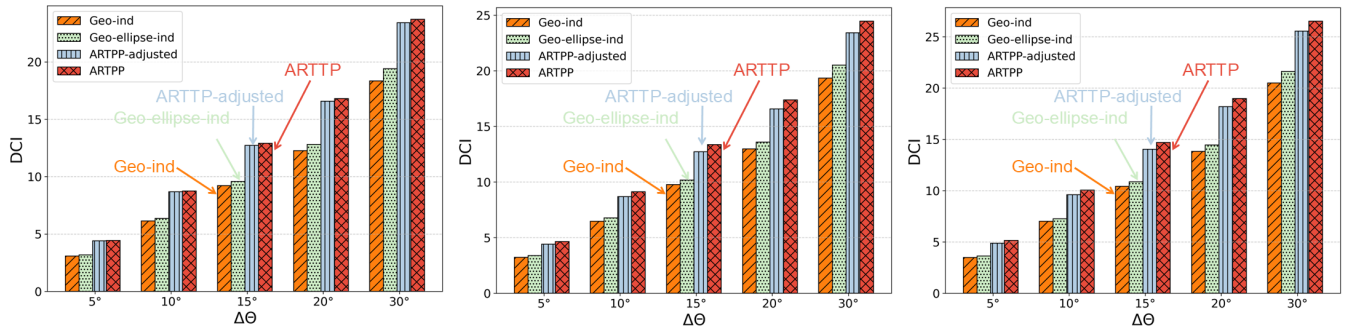
(a) Directionality$_{error}$ (b) Distance$_{error}$

Fig. 10: Comparison of errors with different λ on T-drive dataset.

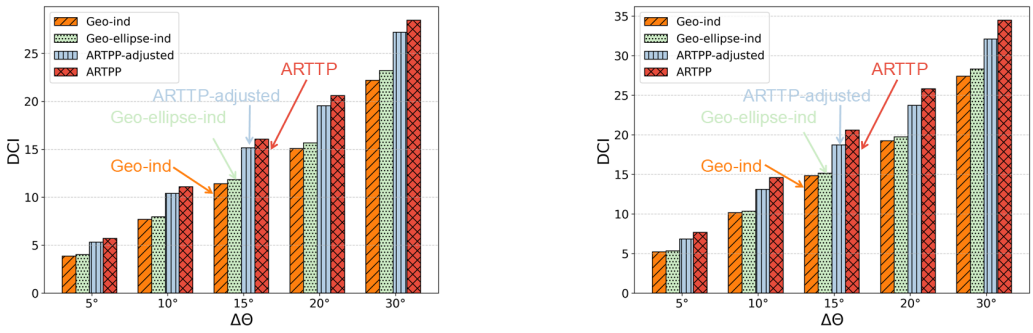
noise distribution, thus providing a better correlation utility at larger values so that when the value is larger, the directional error becomes correspondingly smaller. As ε continues to increase, the directional error becomes progressively smaller, because a larger ε adds less noise, resulting in a smaller directional error. As shown in Fig. 9 (b) and Fig. 10 (b), regardless of ε, the distance error of parameter $\lambda = 0$ is the largest, and the distance error of $\lambda = 1$ is the smallest, and the directionality error is still observed between $\lambda = 1$ and $\lambda = 0.7$ when the adaptive dynamics changes. The phenomenon is due to the non-uniform scaling matrix of the algorithm in the setup of the algorithm, when $\lambda = 0$, the algorithm produces a variance in all directions of 1. And when $\lambda > 0$, the algorithm produces a variance in all directions of the algorithm by the change of position, so when λ is closer to 1, the more information can be retained adjacent to the location of the point on the more information, and so it will result in the reduction of the distance error.

2) Comparison with Other Perturbation Algorithms: Since the Geo-ellipse-ind method cannot be used directly in the scenario of this paper, a sliding window algorithm with a window size of 6 is used to generate noise for this algorithm. For the fairness of the test, all the algorithms are changed to be compared based on the sliding window strategy. The proposed algorithm uses a non-uniform scaling matrix to reflect the effect of the variance of the covariance matrix across the axes, the area of the perturbed noise region of the proposed algorithm is lower than that of the Geo-ind noise when compared to the Geo-ind perturbation mechanism with circular noise distribution. Therefore, to compare the proposed algorithm with other algorithms in a fairer manner, the proposed algorithm will calculate the determinant value of the covariance matrix after the covariance matrix has been generated and adjust the covariance matrix according to this value, so as to ensure that the area of elliptical noise distribution formed by the proposed algorithm is the same as the area of the Geo-ind perturbation mechanism with circular noise distribution. The adjusted algorithm is marked as ARTPP-adjusted.

Fig. 11 and Fig. 12 depict the comparison results of DCI between different methods on the experimental datasets. It can be observed that the proposed algorithms ARTPP and ARTPP-adjusted outperform the other comparative algorithms in terms of the DCI value in both datasets, with Geo-ind performing poorly and ARTPP performing best. This is because the ARTPP algorithm, by focusing on angular variation, ensures that the generated noise will also retain better directionality. In addition, since the effect of the

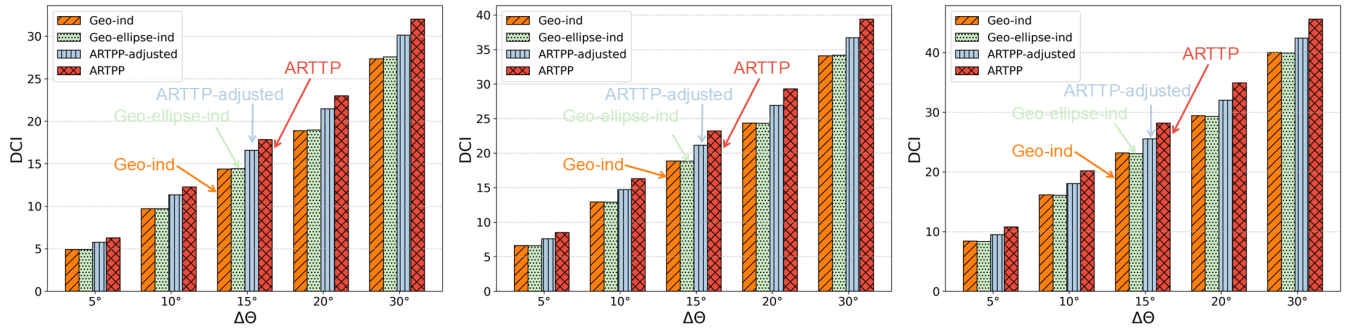


(a) ε = 0.003 (b) ε = 0.005 (c) ε = 0.007

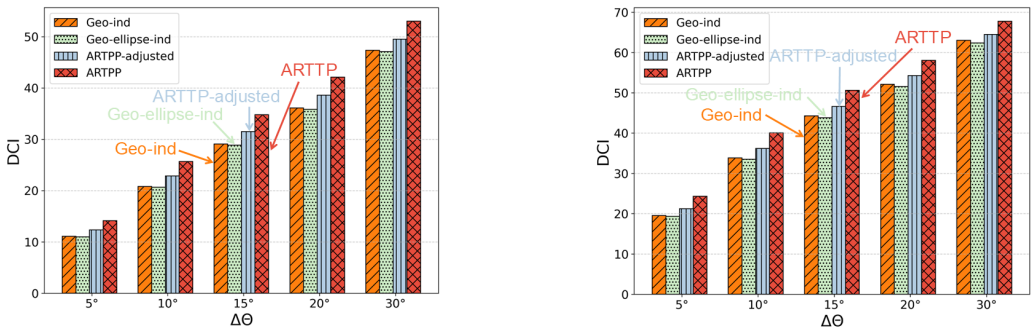


(d) ε = 0.01 (e) ε = 0.02

Fig. 11: Comparison of DCI between different methods on Geolife dataset.



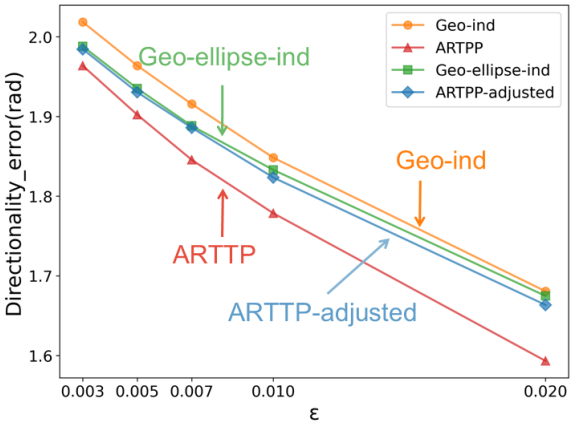
(a) ε = 0.003 (b) ε = 0.005 (c) ε = 0.007



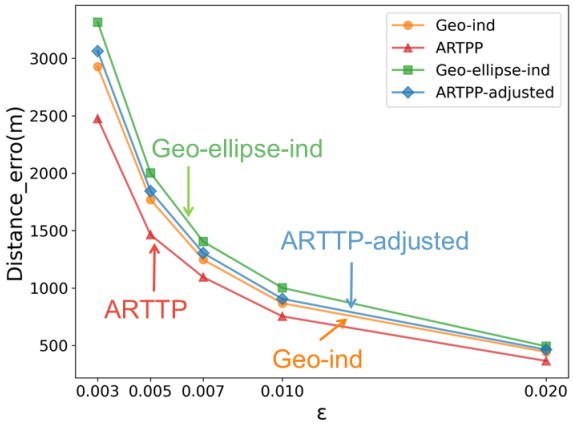
(d) ε = 0.01 (e) ε = 0.02

Fig. 12: Comparison of DCI between different methods on T-drive dataset.

variance of the covariance matrix across all axes is slightly less pronounced than the effect of the variance of Geo-ind, the noise produced by ARTPP is marginally less than that associated with Geo-ind. The ARTPP-adjusted algorithm outperforms both the Geo-ind algorithm and the Geo-ellipse-ind algorithm in terms of DCI metrics, although it performs slightly worse than the original ARTPP algorithm. This is because area adjustment amplifies the amount of noise and therefore causes the ARTPP-adjusted algorithm to perform less well on the DCI metrics. In summary, ARTPP and its adjusted version, ARTPP-adjusted, show better performance in both trajectory datasets.

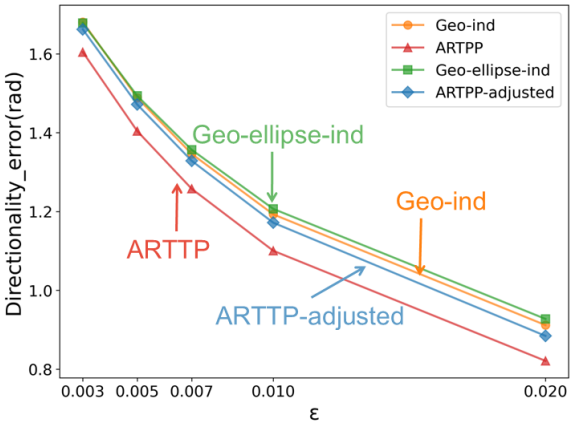


(a) Directionality$_{\text{error}}$

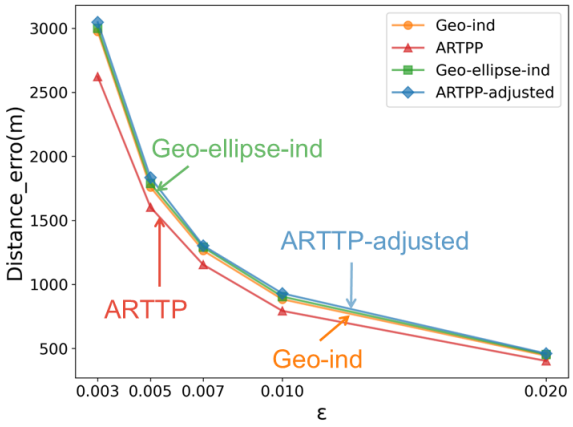


(b) Distance$_{\text{error}}$

Fig. 13: Comparison of errors between different methods on Geolife dataset.



(a) Directionality$_{\text{error}}$



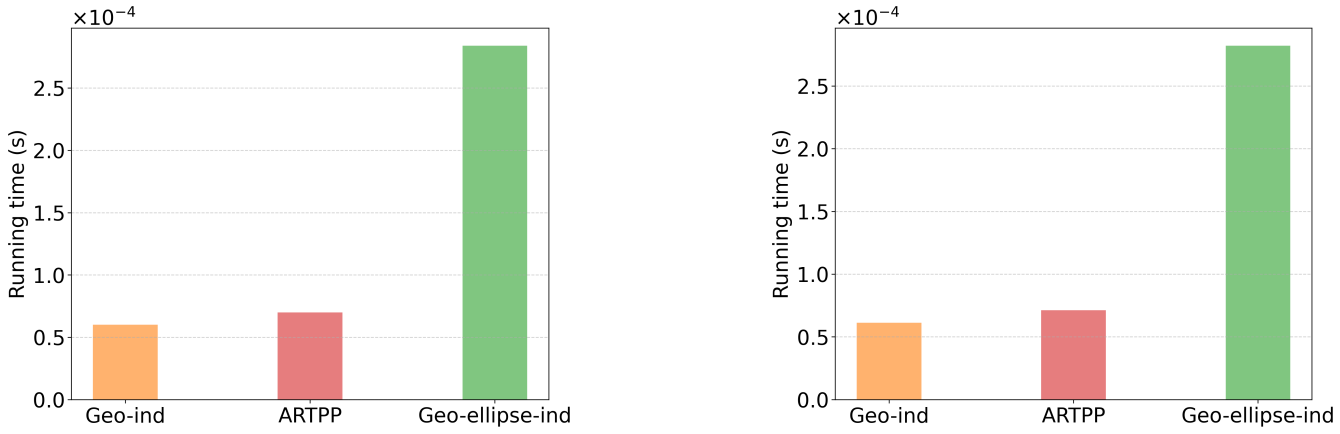
(b) Distance$_{\text{error}}$

Fig. 14: Comparison of errors between different methods on T-drive dataset.

Fig. 13 and Fig. 14 protray the comparison of errors between different methods. It can be seen that the proposed algorithms ARTPP and ARTPP-adjusted outperform the other compared algorithms in terms of directional errors in both datasets, with Geo-ind performing worse and ARTPP performing best. Despite the area adjustment, the ARTPP-adjusted algorithm also still outperforms Geo-ind and Geo-ellipse-ind. As for the distance error metrics, although ARTPP works best, both ARTPP-adjusted and Geo-ellipse-ind errors are larger than Geo-ind, and ARTPP-adjusted is in the middle of Geo-ind and Geo-ellipse-ind. The algorithm ARTPP-adjusted is compared after adjustment and found that the reason why it performs well in directionality metrics is the loss of a certain amount of distance error to achieve it. In geographic location privacy protection, the smaller the perturbation distance represents the better the utility of the algorithm, but with it comes a certain loss of privacy, because too close to the perturbation point is likely to be attacked by the attack so as to carry out the attack, and the larger the perturbation distance represents the better the privacy of the algorithm because of such a location point the attacker is difficult to go to make a judgment, but the cost is sacrificed to the user's utility experience. Therefore, the algorithm ARTPP proposed in this paper is suitable for scenarios that require high relevance, while ARTPP-adjusted is suitable for scenarios that require certain relevance but cannot lose too much privacy.

3) Efficiency Analysis: Since the setting of the privacy budget has no significant effect on the running time of generating perturbed locations, this section analyzes the running time of different perturbation algorithms with ε = 0.003 as an example. We repeat the algorithm 200 times in both datasets and record only the perturbation time in the trajectory each time, and take the average time value as the experimental value.

It can be observed from Fig.15 that the proposed algorithm ARTPP takes slightly longer execution time than the Geo-ind algorithm shorter than the Geo-ellipse-ind. This is because our algorithm spends more time than the Geo-ind algorithm in computing the non-uniform scaling matrix and the isometric transformations matrix, but the computational steps that ARTPP has more than Geo-ind is itself $O(1)$ time complexity, so ARTPP and Geo-ind algorithms are both $O(1)$ time complexity itself. The Geo-ellipse-ind algorithm, on the other hand, has the longest execution time, due to the fact that it requires the computation of the global covariance matrix, a computation that is particularly sensitive to an increase in the amount of data, resulting in a high time overhead.



(a) Running time of different algorithms on Geolife dataset

(b) Running time of different algorithms on T-drive dataset

Fig. 15: Comparison of running time.

It is foreseeable that if the amount of data increases, it will lead to higher time overhead, so the algorithm itself operates with time complexity depending on the number of global covariance matrices. If the number is n, then the algorithm operates with time complexity $O(n)$.

To summarize, our proposed algorithm has more time overhead with Geo-ind algorithm mainly because the algorithm consumes more time utility to improve the utility in correlation, and the Geo-ellipse-ind algorithm results in higher time overhead because of the need to compute the overall covariance matrix.

VI. Conclusion

To address the privacy protection problems of trajectory, metric differential privacy is introduced into the dynamic real-time trajectory scenarios in this paper. The non-uniform scaling matrix and the isometric transformation mapping matrix are designed according to the correlation of neighboring locations. The covariance matrix is constructed by combining the linear transformations of the above two matrices, and on the basis of the perturbation algorithm with elliptical noise distribution is proposed so that the generated perturbation locations can better perceive the correlation changes with the neighboring locations. The significance of the direction change of each location in the trajectory is defined and combined with the proposed perturbation algorithm with elliptical noise distribution. By doing this, more relevant information is retained at locations with high direction change significance, while less relevant information is retained at locations with low direction change significance. A differential privacy metric proof for the proposed algorithm is provided to ensure its privacy. The proposed algorithm is compared and analyzed with other existing correlation algorithms through experiments on real trajectory data, which proves that the proposed algorithm has advantages in the availability of perturbed locations. In the future, we will consider users' personalized needs on this basis and further balance the data utility and the privacy protection effect of this algorithm.

References

[1] H. Huang, X. A. Yao, J. M. Krisp, et al., "Analytics of location-based big data for smart cities: Opportunities, challenges, and future directions," *Computers, Environment and Urban Systems*, vol. 90, p. 101712, 2021.

[2] J. Zhang, F. Xue, X. Cai, et al., "Privacy protection based on many-objective optimization algorithm," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 20, p. e5342, 2019.

[3] C. Costa and M. Y. Santos, "Big Data: State-of-the-art Concepts, Techniques, Technologies, Modeling Approaches and Research Challenges," *IAENG International Journal of Computer Science*, vol. 44, no. 3, pp. 285-301, 2017.

[4] G. K. Pandey, D. S. Gurjar, H. H. Nguyen and S. Yadav, "Security Threats and Mitigation Techniques in UAV Communications: A Comprehensive Survey," *IEEE Access*, vol. 10, pp. 112858-112897, 2022.

[5] M. Shafiq, Z. Gu, O. Cheikhrouhou, et al., "The rise of "Internet of Things": review and open research issues related to detection and prevention of IoT-based security attacks," *Wireless Communications and Mobile Computing*, vol. 2022, p.8669348.

[6] X. Jin, S. Manandhar, K. Kafle, et al., "Understanding IoT security from a market-scale perspective," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 1615-1629.

[7] J. Q. Zhang, X. Wang, Y. F. Yuan, et al., "RcDT: Privacy preservation based on r-constrained dummy trajectory in mobile social networks," *IEEE Access*, vol. 7, pp. 90476-90486, 2019.

[8] W. Zhao, J. Yang, F. Li, C. Pang, J. Li and X. Luo, "Research on Trajectory Clustering Optimization Algorithm Based on Sparse Representation," in *2020 8th International Conference on Digital Home (ICDH)*, 2020, pp. 233-238.

[9] V. A. F. Mina, "On the Privacy Protection of Indoor Location Dataset using Anonymization," *Computers Security*, vol. 117, p. 102665, 2022.

[10] S. Li, H. Tian, H. Shen, Y. Sang, "Privacy-preserving trajectory data publishing by dynamic anonymization with bounded distortion," *ISPRS International Journal of Geo-Information*, vol. 10, no. 2, p. 78, 2021.

[11] C. Chen, W. Lin, S. Zhang, et al., "Personalized trajectory privacy-preserving method based on sensitive attribute generalization and location perturbation," *Intelligent Data Analysis*, vol. 25, no. 5, pp. 1247-1271, 2021.

[12] S. Mahdavifar, F. Deldar, H. Mahdikhani, "Personalized privacy-preserving publication of trajectory data by generalization and distortion of moving points," *Journal of Network and Systems Management*, vol. 30, no. 1, pp. 1-42, 2022.

[13] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming 2006*, pp. 1-12.

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

[15] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, et al., "Broadening the scope of differential privacy using metrics," in *Privacy Enhancing Technologies: 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings*, Springer Berlin Heidelberg, 2013, pp. 82-102.

[16] X. Wu, G. Sun, "A novel dummy-based mechanism to protect privacy on trajectories," in *2014 IEEE International Conference on Data Mining Workshop*, IEEE, 2014, pp. 1120-1125.

[17] S. Shang, L. Chen, Z. Wei, et al., "Trajectory similarity join in spatial networks," *Proceedings of the VLDB Endowment*, vol. 10, no. 11, pp. 1178-1189, 2017.
[18] Z. Hu, J. Yang, J. Zhang, "Trajectory privacy protection method based on the time interval divided," *Computers Security*, vol. 77, pp. 488-499, 2018.
[19] H. Wang, "Trajectory Privacy Protection Mechanism based on Social Attributes," *arXiv preprint arXiv:2212.06600*, 2022.
[20] F. Tian, S. Zhang, L. Lu, et al., "A novel personalized differential privacy mechanism for trajectory data publication," in *2017 International Conference on Networking and Network Applications (NaNA)*, IEEE, 2017, pp. 61-68.
[21] W. Cheng, R. Wen, H. Huang, et al., "OPTDP: Towards optimal personalized trajectory differential privacy for trajectory data publishing," *Neurocomputing*, vol. 472, pp. 201-211, 2022.
[22] X. Sun, Q. Ye, H. Hu, et al., "Synthesizing Realistic Trajectory Data With Differential Privacy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, pp. 5502-5515, 2023.
[23] S. Qiu, D. Pi, Y. Wang, et al., "Novel trajectory privacy protection method against prediction attacks," *Expert Systems with Applications*, vol. 213, p. 118870, 2023.
[24] H. Dai, Z. Wu, S. Wang, and K. Wu, "Grid Anonymous Trajectory Privacy Protection Algorithm Based on Differential Privacy," *IAENG International Journal of Applied Mathematics*, vol. 53, no. 3, pp. 994-1000, 2023.
[25] L. Fan, "Image pixelization with differential privacy," in *Data and Applications Security and Privacy XXXII: 32nd Annual IFIP WG 11.3 Conference, DBSec 2018, Bergamo, Italy, July 16–18, 2018, Proceedings*, 2018, pp. 148-162.
[26] T. Li, C. Clifton, "Differentially private imaging via latent space manipulation," *arXiv preprint arXiv:2103.05472*, 2021.
[27] Y. Han, S. Li, Y. Cao, et al., "Voice-indistinguishability: Protecting voiceprint in privacy-preserving speech data release," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2020, pp. 1-6.
[28] H. Wang, S. Xie, Y. Hong, "VideoDP: A universal platform for video analytics with differential privacy," *arXiv preprint arXiv:1909.08729*, 2019.
[29] Z. Xu, A. Aggarwal, O. Feyisetan, et al., "A differentially private text perturbation method using a regularized mahalanobis metric," *arXiv preprint arXiv:2010.11947*, 2020.
[30] Y. Zhao, D. Yuan, J. T. Du, et al., "Geo-ellipse-indistinguishability: community-aware location privacy protection for directional distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 6957-6967, 2022.
[31] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of National Institute of Sciences of India*, Vol. 12, pp. 49-55, 1936.

Yan Yan is a professor at the School of Computer and Communication, Lanzhou University of Technology, China. She received her Ph.D. degree from Lanzhou University of Technology. Her research interests include, but are not limited to, privacy preserving data publishing, privacy preserving data collection, information hiding and steganalysis, privacy preserving machine learning, and blockchain transaction privacy protection. She is a member of the IEEE and a senior member of the CCF.

Pengbin Yan is currently a postgraduate student at the School of Computer and Communication, Lanzhou University of Technology, China. He received a B.S. degree from Zhengzhou University of Aeronautics with in information management and information systems in 2021. His research interests include location privacy, information security, and geo-information security.

Xinyu Kou is currently a postgraduate student at the School of Computer and Communication, Lanzhou University of Technology, China. She received her B.Eng. degree from Shangluo University in information management and information systems in 2023. Her research interests include location privacy and information security.

Long Wang is currently a postgraduate student at the School of Computer and Communication, Lanzhou University of Technology, China. He received a B.Eng. degree from Henan Finance University in information management and information systems in 2023. His research interests include location privacy, information security, and machine learning.