Computer Standards & Interfaces 97 (2026) 104125

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

AdaTraj-DP: An adaptive privacy framework for context-aware trajectory data publishing[☆]

Yongxin Zhao[a], Chundong Wang[a,b,*], Hao Lin[c,**], Xumeng Wang[d], Yixuan Song[a], Qiuyu Du[c]

[a] Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, Tianjin, China
[b] TianJin Police Institute, Tianjin, China
[c] College of Intelligent Science and Technology (College of Cyberspace Security), Inner Mongolia University of Technology, Inner Mongolia, China
[d] College of Cryptology and Cyber Science, Nankai University, Tianjin, China

ARTICLE INFO

Keywords:
Differential privacy
Trustworthy AI
Trajectory data publishing
Personalized perturbation

ABSTRACT

Trajectory data are widely used in AI-based spatiotemporal analysis but raise privacy concerns due to their fine-grained nature and the potential for individual re-identification. Existing differential privacy (DP) approaches often apply uniform perturbation, which compromises spatial continuity, or adopt personalized mechanisms that overlook structural utility. This study introduces AdaTraj-DP, an adaptive differential privacy framework designed to balance trajectory-level protection and analytical utility. The framework combines context-aware sensitivity detection with hierarchical aggregation. Specifically, a dynamic sensitivity model evaluates privacy risks according to spatial density and semantic context, enabling adaptive allocation of privacy budgets. An adaptive perturbation mechanism then injects noise proportionally to the estimated sensitivity and represents trajectories through Hilbert-based encoding for prefix-oriented hierarchical aggregation with layer-wise budget distribution. Experiments conducted on the T-Drive and GeoLife datasets indicate that AdaTraj-DP maintains stable query accuracy, spatial consistency, and downstream analytical utility across varying privacy budgets while satisfying formal differential privacy guarantees.

1. Introduction

The proliferation of mobile devices, GPS sensors, and intelligent transportation infrastructures has resulted in the large-scale collection of spatiotemporal data. Such data serve as the foundation for numerous Location-Based Services (LBS), including navigation, ride-hailing, and urban planning [1,2]. Trajectory datasets record detailed sequences of individual movements, enabling a wide range of AI applications such as traffic forecasting, mobility prediction, and behavioral modeling. These applications have become indispensable for smart city management and autonomous systems, where the integrity and granularity of trajectory data directly affect analytical and decision-making accuracy.

Despite their utility, trajectory datasets raise critical privacy concerns for trustworthy AI. A single trajectory may expose an individual's home, workplace, or health-related locations, revealing sensitive behavioral patterns and social relationships [3,4]. Even after removing explicit identifiers, re-identification attacks can reconstruct personal traces with minimal auxiliary information [5]. Consequently, ensuring differential privacy for trajectory data has become essential to support reliable and ethically compliant AI development.

Differential Privacy (DP) [6] provides a rigorous mathematical guarantee against information leakage. However, its application to trajectory publishing introduces a persistent trade-off between privacy strength, data utility, and personalization, which conventional mechanisms fail to reconcile. Two primary gaps remain unresolved: (1) the tension between point-level perturbation and structural integrity;(2) the difficulty of adapting privacy budgets to varying contextual sensitivity. Early studies injected uniform Laplace noise into each location point [7,8], which protected individual coordinates but severely distorted the spatiotemporal correlation essential for route-level analysis. Subsequent hierarchical schemes based on prefix trees or space-filling curves [9,10] preserved aggregate statistics but relied on global, fixed privacy parameters, ignoring heterogeneous sensitivity across trajectories. Recent progress in Personalized Differential Privacy (PDP) [11–13] introduced adaptive noise based on semantic or frequency-based sensitivity, yet these methods typically lack integration with hierarchical

[☆] This article is part of a Special issue entitled: 'Secure AI' published in Computer Standards & Interfaces.
[*] Corresponding author at: Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, Tianjin, China.
[**] Corresponding author.

E-mail addresses: zyx4237@163.com (Y. Zhao), michael3769@163.com (C. Wang), suzukaze_aoba@126.com (H. Lin), wangxumeng@nankai.edu.cn (X. Wang), fykatb0824@163.com (Q. Du).

https://doi.org/10.1016/j.csi.2025.104125
Received 29 October 2025; Received in revised form 25 December 2025; Accepted 29 December 2025
Available online 30 December 2025

aggregation, resulting in limited query accuracy and poor scalability for AI model training.

To bridge this gap, we propose AdaTraj-DP, an adaptive differentially private trajectory publishing framework that unifies context-aware sensitivity modeling and hierarchical aggregation. AdaTraj-DP introduces a two-stage protection mechanism. The first stage detects and quantifies sensitivity using contextual and statistical cues, allowing adaptive privacy budget assignment at the point level. The second stage encodes perturbed trajectories into a hierarchical prefix tree, applying layer-wise budget allocation to preserve structural consistency for downstream analysis. This design ensures both localized protection and global analytical utility, addressing the core limitations of prior DP-based trajectory mechanisms.

The main contributions of this work are summarized as follows:

1. We propose AdaTraj-DP, an adaptive framework that unifies personalized perturbation and hierarchical aggregation. By establishing a mathematical link between local coordinate noise and global prefix-tree structures, the framework ensures that fine-grained point-level protection remains structurally consistent with trajectory-level differential privacy guarantees, enabling high-fidelity reconstruction for downstream tasks.
2. We design a context-aware sensitivity model that combines spatial density with semantic context to guide adaptive budget allocation. This mechanism quantifies privacy risks at a granular level, enabling the dynamic adjustment of perturbation intensity to balance privacy protection and data fidelity.
3. We implement a hierarchical aggregation scheme utilizing Hilbert spatial mapping and logarithmic layer-wise budget distribution. Experiments on the T-Drive and GeoLife datasets validate the framework's effectiveness in preserving query accuracy, spatial consistency, and AI model performance under varying privacy budgets.

2. Related work

Existing privacy-preserving trajectory publishing approaches can be broadly categorized into three classes: (1) foundational differential privacy models that ensure privacy but compromise trajectory continuity; (2) structural aggregation mechanisms that enhance data utility via hierarchical organization; and (3) personalized and adaptive privacy protection strategies that tailor noise to sensitivity but often lack integration with structural models. This section reviews these three directions and discusses recent advances that motivate AdaTraj-DP.

2.1. Foundational models for differentially private trajectory publishing

Differential Privacy (DP) [6] is the standard formalism for privacy-preserving data publication. Early approaches discretize continuous spatio-temporal domains and inject Laplace noise into cell counts or simple aggregates [14,15], but such methods often disrupt trajectory continuity and reduce utility for route-level analysis [7]. To address this, research has explored trajectory generalization and synthetic data generation under DP, including clustering-based generalization [16] and GAN-based synthetic trajectory models [17–19]. Work on DP-aware data exploration and visualization—e.g., DPKnob and Defogger—highlights the challenge of configuring DP mechanisms to balance utility and risk in interactive settings and motivates user- or task-guided privacy configuration [20,21].

2.2. Structural aggregation for utility enhancement

Hierarchical structures—such as prefix trees, Hilbert-encoded sequences, and spatial index trees—have been widely adopted to preserve aggregate query utility under DP. Early prefix-tree methods aggregate shared prefixes to reduce noise impact [22,23], while R-tree and quadtree variants support spatial indexing under privacy constraints [7,10]. Recent work improves spatial locality and query accuracy using Hilbert/Geohash encodings and adaptive tree strategies [9]. Zhao et al.'s PerTrajTree-DP further integrates point-level sensitivity with prefix-tree publishing to better support trustworthy AI analytics [24]. Complementary systems research on private data access and explanation (e.g., DPXPlain, Saibot) demonstrates practical techniques for supporting DP-protected analytics and helping users interpret noisy aggregates [25,26].

2.3. Personalized and adaptive privacy protection

Personalized Differential Privacy (PDP) methods adapt protection to varying point- or user-level sensitivity. Semantics-driven approaches use POI categories or external labels to identify sensitive locations [27,28], and movement-model-based frameworks like OPTDP estimate privacy risk from mobility patterns [11]. Statistical personalization methods infer sensitivity from dataset properties; for example, TF–IDF-based approaches quantify local importance and global rarity to guide budget allocation [12,13]. Interactive tools and visual analytics (DPKnob, Defogger) provide practical support for configuring heterogeneous DP strategies according to utility goals [20,21].

In parallel, recent advances in differentially private deep learning and private model training yield methods for improved utility in noisy training regimes (e.g., optimized DP-SGD variants, selective-update training, and heterogeneous-noise schemes) that can inform budget allocation and model-aware privacy strategies in trajectory publishing [25,26,29–31]. These works highlight opportunities to close the gap between personalized point-level protection and structural aggregation, motivating AdaTraj-DP's integration of context-aware sensitivity detection, adaptive perturbation, and hierarchical encoding to support AI-oriented downstream tasks.

3. Preliminaries

Trajectory Representation. A trajectory T_i of user u_i is a temporally ordered sequence of geo-referenced points [32]:

$$T_i = \{(p_{i,1}, t_{i,1}), (p_{i,2}, t_{i,2}), \dots, (p_{i,L_i}, t_{i,L_i})\}, \tag{1}$$

where $p_{i,j} = (\mathrm{lat}_{i,j}, \mathrm{lon}_{i,j})$ denotes the spatial coordinate and $t_{i,j}$ is the timestamp. The trajectory dataset is denoted as $\mathcal{D} = \{T_1, T_2, \dots, T_N\}$.

Each point can be projected into a discrete grid cell $c_{i,j}$ for statistical analysis or further spatial encoding. The dimensionality and sampling irregularity of $\mathcal{D}$ result in high sparsity and heterogeneous sensitivity among locations, which requires adaptive privacy mechanisms.

Differential Privacy. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two neighboring datasets differing in at most one trajectory. A randomized mechanism $\mathcal{M}$ satisfies ε-differential privacy if for any measurable subset O in the output space:

$$\Pr[\mathcal{M}(\mathcal{D}_1) \in O] \le e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{D}_2) \in O]. \tag{2}$$

The privacy budget $\varepsilon > 0$ controls the trade-off between privacy protection and data utility. Smaller ε implies stronger privacy guarantees but larger perturbation noise.

For a numerical query $f : \mathcal{D} \to \mathbb{R}^k$ with ℓ_1 sensitivity $\Delta f = \max_{\mathcal{D}_1,\mathcal{D}_2} \|f(\mathcal{D}_1) - f(\mathcal{D}_2)\|_1$, the Laplace mechanism adds independent noise drawn from the Laplace distribution:

$$\mathcal{M}(\mathcal{D}) = f(\mathcal{D}) + \mathrm{Lap}(\Delta f/\varepsilon). \tag{3}$$

This mechanism provides ε-differential privacy and is used in subsequent trajectory perturbation and aggregation processes.

Geographic Indistinguishability. For any two spatial points $x, x' \in \mathbb{R}^2$ and any reported location z, a mechanism $\mathcal{K}$ achieves ε-geographic indistinguishability if

$$\Pr[\mathcal{K}(x) = z] \le e^{\varepsilon \cdot d(x,x')} \Pr[\mathcal{K}(x') = z], \tag{4}$$

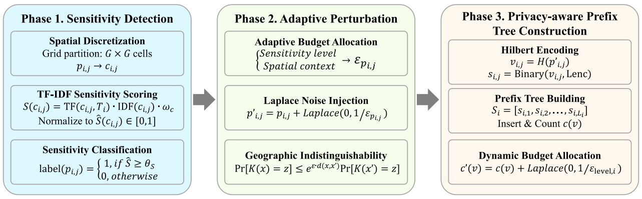

Phase 1. Sensitivity Detection
Spatial Discretization
Grid partition: $G \times G$ cells
$p_{i,j} \rightarrow c_{i,j}$
TF-IDF Sensitivity Scoring
$S(c_{i,j}) = \mathrm{TF}(c_{i,j}, T_i) \cdot \mathrm{IDF}(c_{i,j}) \cdot \omega_c$
Normalize to $\hat{S}(c_{i,j}) \in [0,1]$
Sensitivity Classification
$\mathrm{label}(p_{i,j}) = \begin{cases} 1, if\ \hat{S} \geq \theta_S \\ 0, otherwise \end{cases}$
Phase 2. Adaptive Perturbation
Adaptive Budget Allocation
$\begin{cases} Sensitivity\ level \\ Spatial\ context \end{cases} \rightarrow \varepsilon_{p_{i,j}}$
Laplace Noise Injection
$p'_{i,j} = p_{i,j} + Laplace(0, 1/\varepsilon_{p_{i,j}})$
Geographic Indistinguishability
$\Pr[K(x) = z] \leq e^{\varepsilon \cdot d(x,x')} \Pr[K(x') = z]$
Phase 3. Privacy-aware Prefix Tree Construction
Hilbert Encoding
$v_{i,j} = H(p'_{i,j})$
$s_{i,j} = \mathrm{Binary}(v_{i,j}, \mathrm{Lenc})$
Prefix Tree Building
$S_i = [s_{i,1}, s_{i,2}, \ldots, s_{i,L_i}]$
Insert & Count $c(v)$
Dynamic Budget Allocation
$c'(v) = c(v) + Laplace(0, 1/\varepsilon_{\mathrm{level},i})$

Fig. 1. Framework of the proposed AdaTraj-DP scheme.

where $d(x, x')$ is the Euclidean distance between x and x' [33].

This formulation extends differential privacy to continuous spatial domains and provides distance-dependent protection.

Hierarchical Aggregation Structure. Trajectory data exhibit hierarchical correlations that can be represented through prefix-based aggregation. Let each discretized or encoded trajectory be expressed as a sequence of spatial identifiers $S_i = [s_{i,1}, s_{i,2}, \ldots, s_{i,L_i}]$. A prefix tree $\mathcal{T}$ organizes all trajectories in $\mathcal{D}$ by shared prefixes, where each node v corresponds to a spatial prefix and maintains a count $c(v)$ of trajectories passing through it. The hierarchical form allows noise to be injected at multiple granularities while preserving global spatial consistency.

The total privacy budget $\varepsilon_{\text{tree}}$ is distributed across tree layers to balance upper-level accuracy and lower-level detail preservation.

Problem Definition. Given a trajectory dataset $\mathcal{D}$ consisting of N users and a total privacy budget$\varepsilon_{\text{total}}$, the objective is to design a mechanism $\mathcal{M}_{\text{traj}}$ that releases a trajectory dataset $\tilde{\mathcal{D}} = \mathcal{M}_{\text{traj}}(\mathcal{D})$ satisfying:

(1) $\mathcal{M}_{\text{traj}}$ ensures $\varepsilon_{\text{total}}$-differential privacy at the trajectory level;
(2) The released dataset $\tilde{\mathcal{D}}$ preserves statistical and structural properties essential for AI-based spatiotemporal analysis;
(3) The expected analytical error between results obtained from $\tilde{\mathcal{D}}$ and $\mathcal{D}$ remains bounded.

Let $f_{\text{AI}}(\cdot)$ denote an AI model trained or evaluated on trajectory data. The utility preservation objective is formulated as

$$L_{\text{utility}} = \mathbb{E}\left[\| f_{\text{AI}}(\tilde{\mathcal{D}}) - f_{\text{AI}}(\mathcal{D}) \|_2^2\right], \tag{5}$$

subject to $\tilde{\mathcal{D}}$ satisfying $\varepsilon_{\text{total}}$-differential privacy. The goal is to minimize L_{utility} while maintaining formal privacy guarantees.

4. Proposed framework

Rapid development of AI-driven spatiotemporal analysis has increased the demand for high-quality trajectory data with strong privacy protection. Traditional differential privacy mechanisms often adopt fixed noise scales or uniform budget allocation, which can cause excessive utility degradation in dense areas or insufficient protection in sensitive regions. To address these limitations, this study proposes **AdaTraj-DP**, a framework that integrates adaptive personalized perturbation with hierarchical aggregation to achieve trajectory-level differential privacy while maintaining analytical utility for AI-based modeling. As illustrated in Fig. 1, AdaTraj-DP operates in three main phases: (1) trajectory preprocessing and context-aware sensitivity detection; (2) adaptive personalized perturbation guided by local sensitivity and spatial density; (3) hierarchical aggregation using Hilbert encoding and dynamic layer-wise budget allocation.

4.1. Context-aware sensitivity detection

Let $\mathcal{D} = \{T_1, \ldots, T_N\}$ denote the trajectory dataset after basic preprocessing. Each trajectory $T_i = \{(p_{i,1}, t_{i,1}), \ldots, (p_{i,L_i}, t_{i,L_i})\}$ consists of temporally ordered spatial points $p_{i,j} = (\text{lat}_{i,j}, \text{lon}_{i,j})$. The objective of this phase is to quantify the privacy sensitivity of each spatial point by combining statistical frequency and contextual semantics to guide subsequent adaptive perturbation.

Spatial Discretization. The continuous geographical domain is partitioned into a uniform grid of $G \times G$ cells. Each point $p_{i,j}$ is mapped to a corresponding grid cell $c_{i,j}$. This transformation converts raw coordinates into discrete spatial tokens, enabling frequency-based statistical analysis.

Context-aware Sensitivity Measure. For each cell $c_{i,j}$, a sensitivity score $S(c_{i,j})$ is defined as

$$S(c_{i,j}) = \text{TF}(c_{i,j}, T_i) \cdot \text{IDF}(c_{i,j}) \cdot \omega_c, \tag{6}$$

where $\text{TF}(c_{i,j}, T_i) = \frac{\text{count}(c_{i,j} \in T_i)}{L_i}$ represents the normalized local frequency of visits within trajectory T_i, and $\text{IDF}(c_{i,j}) = \log \frac{|\mathcal{D}|}{|\{T_k \in \mathcal{D} : c_{i,j} \in T_k\}|}$ denotes the global rarity of the location across the dataset. The term ω_c is a contextual weighting coefficient that quantifies the semantic sensitivity of a location category. Following the semantic sensitivity hierarchy established in [34], we assign higher weights to privacy-critical categories (e.g., $\omega_{healthcare} = 1.5, \omega_{residential} = 1.2$) to enforce stricter protection, while assigning lower base weights to public infrastructure (e.g., $\omega_{road} = 1.0$). These semantic categories are mapped from public map services (e.g., OpenStreetMap), ensuring that the sensitivity configuration relies solely on public knowledge and does not consume the private budget.

Normalization and Classification. To unify the sensitivity scale, all scores are normalized into $[0, 1]$:

$$\hat{S}(c_{i,j}) = \frac{S(c_{i,j}) - \min(S)}{\max(S) - \min(S)}. \tag{7}$$

Each point $p_{i,j}$ is then labeled as sensitive or non-sensitive according to a predefined threshold θ_S:

$$\text{label}(p_{i,j}) = \begin{cases} 1, & \text{if } \hat{S}(c_{i,j}) \geq \theta_S, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The resulting annotated dataset is represented as $\mathcal{D}' = \{T'_1, T'_2, \ldots, T'_N\}$, where each T'_i contains the points and corresponding sensitivity labels. The normalized score $\hat{S}(c_{i,j})$ serves as a continuous privacy indicator in the subsequent adaptive perturbation phase.

4.2. Adaptive personalized perturbation

This phase injects controlled noise into all trajectory points in $\mathcal{D}'$ to ensure trajectory-level differential privacy. All locations are perturbed to avoid inference risks arising from selective protection. The perturbation strength is adaptively adjusted based on the normalized sensitivity $\hat{S}(c_{i,j})$ and local spatial density, allowing the mechanism to preserve analytical fidelity while maintaining formal privacy guarantees.

Adaptive Privacy Budget Allocation. Each trajectory point $p_{i,j}$ is assigned an individual privacy budget $\varepsilon_{p_{i,j}}$ determined by both its sensitivity level and spatial context.

Let $\rho(p_{i,j})$ denote the local point density around $p_{i,j}$ within a neighborhood radius r. The adaptive budget is defined as

$$\varepsilon_{p_{i,j}} = \varepsilon_{\max} - (\varepsilon_{\max} - \varepsilon_{\min}) \times \left(\alpha \hat{S}(c_{i,j}) + (1 - \alpha)(1 - \rho(p_{i,j}))\right), \tag{9}$$

where $\alpha \in [0, 1]$ controls the balance between sensitivity-based and density-based adaptation.

A higher $\hat{S}(c_{i,j})$ or lower $\rho(p_{i,j})$ leads to a smaller $\varepsilon_{p_{i,j}}$, introducing stronger noise for privacy-critical or sparsely visited regions. The range $[\varepsilon_{\min}, \varepsilon_{\max}]$ defines the permissible privacy strength, ensuring stability across heterogeneous data distributions.

Two-Dimensional Laplace Perturbation. For each point $p_{i,j} = (\text{lat}_{i,j}, \text{lon}_{i,j})$, independent Laplace noise is applied to both coordinates according to the assigned privacy budget:

$$p'_{i,j} = \begin{cases} \text{lat}_{i,j} + \text{Laplace}(0, 1/\varepsilon_{p_{i,j}}) \\ \text{lon}_{i,j} + \text{Laplace}(0, 1/\varepsilon_{p_{i,j}}) \end{cases} \tag{10}$$

Algorithm 1 Adaptive Personalized Perturbation under AdaTraj-DP

```
Input: Annotated dataset $\mathcal{D}'$, privacy range $[\varepsilon_{\min}, \varepsilon_{\max}]$, sensitivity
    scores $\hat{S}$, balance coefficient $\alpha$
Output: Perturbed dataset $\mathcal{D}''$
$\mathcal{D}'' \leftarrow \emptyset$
for each trajectory $T_i \in \mathcal{D}'$ do
    $T_i'' \leftarrow \emptyset$
    for each point $p_{i,j}$ in $T_i$ do
        Compute local density $\rho(p_{i,j})$
        $\varepsilon_{p_{i,j}} \leftarrow \varepsilon_{\max} - (\varepsilon_{\max} - \varepsilon_{\min}) \times (\alpha \hat{S}(c_{i,j}) + (1-\alpha)(1-\rho(p_{i,j})))$
        $n_{\text{lat}} \sim \text{Laplace}(0, 1/\varepsilon_{p_{i,j}})$
        $n_{\text{lon}} \sim \text{Laplace}(0, 1/\varepsilon_{p_{i,j}})$
        $p'_{i,j} \leftarrow (\text{lat}_{i,j} + n_{\text{lat}}, \text{lon}_{i,j} + n_{\text{lon}})$
        Append $p'_{i,j}$ to $T_i''$
    end for
    Add $T_i''$ to $\mathcal{D}''$
end for
return $\mathcal{D}''$
```

Algorithm 2 Dynamic Hierarchical Aggregation under AdaTraj-DP

```
Input: Perturbed dataset $\mathcal{D}''$, total tree budget $\varepsilon_{\text{tree}}$, height $h$,
    parameters $a$, $\gamma$, encoding length $L_{\text{enc}}$
Output: Privacy-aware prefix tree $\mathcal{T}'$
Initialize empty tree $\mathcal{T}$
for each trajectory $T_i'' = \{p'_{i,1}, \dots, p'_{i,L_i}\}$ in $\mathcal{D}''$ do
    Encode trajectory:
    $S_i \leftarrow [\text{Encode1D}(H(p'_{i,1})), \dots, \text{Encode1D}(H(p'_{i,L_i}))]$
    Insert $S_i$ into $\mathcal{T}$ and increment node counts along each path
end for
for layer $i = 1$ to $h$ do
    Compute node count variance $\sigma_i^2$
    $\varepsilon_{\text{level},i} \leftarrow \frac{(\log(i+a))(1+\gamma\sigma_i^2)}{\sum_{j=1}^{h}(\log(j+a))(1+\gamma\sigma_j^2)} \cdot \varepsilon_{\text{tree}}$
    for each node $v$ at layer $i$ do
        $c'(v) \leftarrow c(v) + \text{Laplace}(0, 1/\varepsilon_{\text{level},i})$
        Update $c(v) \leftarrow c'(v)$
    end for
end for
return $\mathcal{T}'$
```

The perturbed trajectory $T_i'' = \{p'_{i,1}, p'_{i,2}, \dots, p'_{i,L_i}\}$ is constructed by replacing each original point with its perturbed counterpart. The complete differentially private dataset is denoted as $\mathcal{D}'' = \{T_1'', T_2'', \dots, T_N''\}$.

Algorithm 1 outlines the adaptive personalized perturbation procedure.

4.3. Hierarchical aggregation with dynamic budget allocation

This phase organizes the perturbed trajectories into a structured form for privacy-preserving analytical querying and AI model training. A hierarchical prefix tree is constructed from the encoded trajectories, where node counts are perturbed under a dynamically adjusted budget to preserve global consistency while mitigating noise propagation.

Spatial Encoding via Hilbert Curve. Each perturbed point $p'_{i,j} \in \mathcal{D}''$ is mapped into a one-dimensional integer value $v_{i,j}$ using a Hilbert space-filling curve $H(\cdot)$, ensuring spatial locality preservation:

$$v_{i,j} = H(p'_{i,j}). \tag{11}$$

Each integer value $v_{i,j}$ is then converted into a fixed-length binary string $s_{i,j}$ of length L_{enc}, forming a discretized trajectory representation $S_i = [s_{i,1}, s_{i,2}, \dots, s_{i,L_i}]$. The set of all encoded trajectories $\{S_i\}$ constitutes the input to hierarchical aggregation. The technical details of this Hilbert-to-binary-string encoding, including the relationship between the curve's order and the string length, are elaborated in Appendix.

Prefix Tree Construction. A prefix tree $\mathcal{T}$ is built from $\{S_i\}$, where each path from the root to a node v represents a spatial prefix, and the node count $c(v)$ indicates the number of trajectories sharing that prefix. The maximum tree depth h corresponds to the maximum trajectory length or encoding depth.

Dynamic Layer-wise Budget Allocation. The total privacy budget $\varepsilon_{\text{tree}}$ is distributed across tree layers according to both layer depth and statistical variance. Let σ_i^2 denote the empirical variance of node counts at layer i. The adaptive allocation for layer i is defined as

$$\varepsilon_{\text{level},i} = \frac{(\log(i+a)) \cdot (1+\gamma\sigma_i^2)}{\sum_{j=1}^{h}(\log(j+a))(1+\gamma\sigma_j^2)} \cdot \varepsilon_{\text{tree}}, \tag{12}$$

where $a > 0$ is a smoothing parameter and $\gamma \geq 0$ controls the weight of variance-based adjustment. Adopting the logarithmic strategy from [9], the function $\log(i + a)$ is selected to smooth the budget decay across layers. Unlike linear or exponential allocation schemes, which might excessively penalize deeper layers and lead to significant information loss in fine-grained trajectories, the logarithmic term ensures that leaf nodes retain sufficient privacy budget to preserve local spatial details.

Differentially Private Node Perturbation. For each node v at layer i, the sensitivity of its count query is $\Delta f = 1$. Laplace noise is applied according to its layer-wise budget:

$$c'(v) = c(v) + \text{Laplace}\left(0, \frac{1}{\varepsilon_{\text{level},i}}\right). \tag{13}$$

The resulting prefix tree $\mathcal{T}'$ with perturbed counts serves as a privacy-preserving hierarchical representation supporting aggregate analytics and AI-based trajectory modeling.

Algorithm 2 summarizes the hierarchical aggregation process with dynamic budget adjustment.

4.4. Privacy analysis

The proposed AdaTraj-DP framework comprises two sequential privacy-preserving mechanisms: adaptive personalized perturbation (with budget $\varepsilon_{\text{point}}$) and hierarchical aggregation (with budget $\varepsilon_{\text{tree}}$). By the sequential composition theorem of differential privacy, the total privacy guarantee satisfies

$$\varepsilon_{\text{total}} = \varepsilon_{\text{point}} + \varepsilon_{\text{tree}}. \tag{14}$$

Privacy of Adaptive Personalized Perturbation ($\varepsilon_{\text{point}}$). The adaptive perturbation mechanism assigns an individual privacy budget $\varepsilon_{p_{i,j}}$ to each trajectory point $p_{i,j}$ derived from its normalized sensitivity $\hat{S}(c_{i,j})$ and local density $\rho(p_{i,j})$. To ensure rigorous privacy guarantees, it is assumed that the global weighting parameters (e.g., contextual weights ω_c and density thresholds) are computed from public sources, such as map topologies or non-sensitive historical statistics. This reliance on public metadata is a standard practice in privacy-preserving spatial publishing [14,33], ensuring that the sensitivity calibration process itself does not leak private information. Consequently, the allocated budget $\varepsilon_{p_{i,j}}$ depends solely on the characteristics of its corresponding trajectory T_i. Under this assumption:

(1) The assignment of $\varepsilon_{p_{i,j}}$ relies solely on local statistics within T_i and public constants, which ensures independence among users.
(2) Each trajectory is processed through an independent Laplace mechanism. For any point $p_{i,j}$, the Laplace mechanism with scale $1/\varepsilon_{p_{i,j}}$ satisfies $\varepsilon_{p_{i,j}}$-differential privacy.

(3) Because the budgets are bounded within $[\varepsilon_{\min}, \varepsilon_{\max}]$, the overall privacy cost of this phase is dominated by the smallest allocated budget, and the worst-case (strongest) guarantee corresponds to $\varepsilon_{\min}$-DP for each point.
(4) By parallel composition across trajectories, the global privacy consumption of this phase is $\varepsilon_{\text{point}} = \varepsilon_{\max}$, representing the maximum privacy loss incurred when the weakest noise is added.

Hence, the adaptive perturbation phase satisfies $\varepsilon_{\max}$-differential privacy.

Privacy of Hierarchical Aggregation ($\varepsilon_{\text{tree}}$). The hierarchical aggregation mechanism constructs a prefix tree and perturbs its node counts with layer-specific noise calibrated by $\varepsilon_{\text{level},i}$. Each trajectory affects exactly one node per layer, implying that the sensitivity of the count query at any layer is $\Delta f = 1$. Adding Laplace noise with scale $1/\varepsilon_{\text{level},i}$ guarantees $\varepsilon_{\text{level},i}$-DP for that layer.

Because the per-layer budgets $\varepsilon_{\text{level},i}$ are partitioned from $\varepsilon_{\text{tree}}$ according to

$$\sum_{i=1}^{h} \varepsilon_{\text{level},i} = \varepsilon_{\text{tree}}, \tag{15}$$

and the layers are sequentially composed along each trajectory path, the entire prefix tree synthesis mechanism satisfies $\varepsilon_{\text{tree}}$-differential privacy. The dynamic allocation factor $(1 + \gamma\sigma_i^2)$ modifies the budget distribution without altering the total privacy bound, ensuring that the overall guarantee remains unchanged.

Overall Privacy Guarantee. Applying the sequential composition theorem to the two phases yields the total privacy protection level:

$$\varepsilon_{\text{total}} = \varepsilon_{\max} + \varepsilon_{\text{tree}}. \tag{16}$$

This ensures that AdaTraj-DP provides formal, trajectory-level differential privacy. The adaptive and hierarchical mechanisms jointly maintain consistent privacy guarantees while supporting utility-preserving analysis for AI-based spatiotemporal modeling.

5. Experimental evaluation

This section presents an extensive empirical evaluation of the proposed AdaTraj-DP framework. The experiments aim to validate both privacy preservation and analytical utility in AI-oriented trajectory publishing. Specifically, we address the following research questions:

- **RQ1:** How does the total privacy budget $\varepsilon_{\text{total}}$ affect the analytical utility of the released trajectories?
- **RQ2:** How does AdaTraj-DP perform compared to state-of-the-art differential privacy mechanisms in terms of accuracy and computational efficiency?
- **RQ3:** What are the impacts of the adaptive parameters—including allocation ratio α and variance factor γ—on privacy–utility trade-offs?

5.1. Experimental setup

This subsection introduces the datasets, baseline methods, evaluation metrics, and parameter configurations used in the experiments.

5.1.1. Datasets

Experiments are primarily conducted on the widely used T-Drive dataset, which records GPS trajectories of 10,357 taxis in Beijing over seven days (February 2–8, 2008) [35]. It contains approximately 15 million spatial points after preprocessing. To further verify cross-domain robustness, we additionally include the GeoLife dataset [36], which comprises 17,621 trajectories from 182 users, covering both dense urban and sparse suburban mobility patterns.

Both datasets are preprocessed by: (1) removing sampling intervals exceeding 300 s; (2) filtering out trajectories shorter than 20 points; (3) normalizing all coordinates into a $[0, 1] \times [0, 1]$ grid to ensure scale comparability.

These datasets collectively provide both high-density and low-density spatial distributions, enabling a fair evaluation of the proposed context-aware sensitivity modeling.

5.1.2. Baseline methods

To demonstrate the advantages of AdaTraj-DP, we compare it with four representative baselines, each reflecting a distinct privacy design paradigm:

- **HA-Tree** [9]: A hierarchical aggregation method based on Hilbert mapping and fixed logarithmic budget allocation, representing state-of-the-art static DP trees.
- **TFIDF-DP** [13]: A personalized perturbation method using TF–IDF-based sensitivity scoring without hierarchical structure, corresponding to point-level DP only.
- **QJLP (LDP)** [7]: A local differential privacy baseline where each trajectory is perturbed independently on the client side.
- **AdaTraj-DP (Ours)**: The proposed adaptive framework that combines context-aware sensitivity detection, adaptive perturbation, and dynamic hierarchical aggregation.

5.1.3. Evaluation metrics

Performance is evaluated from three complementary perspectives:

Data Utility. We adopt three quantitative metrics: Mean Absolute Error (MAE), Mean Relative Error (MRE), and Hausdorff Distance (HD). MAE and MRE evaluate accuracy for range-count queries on perturbed trajectories, while HD measures spatial fidelity between original and released datasets.

Model Utility. To align with AI-oriented evaluation, we train a downstream trajectory classification model based on a lightweight Mamba encoder [37]. The model predicts driver ID from trajectory segments, and classification accuracy on the perturbed data reflects end-task utility (U_{cls}).

Computational Efficiency. We report total runtime (T_{total}) from preprocessing to privacy-protected publication, including all three phases of AdaTraj-DP.

5.1.4. Parameter configuration

Unless otherwise stated, experiments use the following default configuration: the total privacy budget $\varepsilon_{\text{total}}$ is divided by an allocation ratio α, where $\alpha \in [0.3, 0.7]$ controls the portion used for adaptive perturbation ($\varepsilon_{\text{point}}$), and $(1-\alpha)$ for hierarchical aggregation ($\varepsilon_{\text{tree}}$):

$$\varepsilon_{\text{point}} = \alpha\varepsilon_{\text{total}}, \;\; \varepsilon_{\text{tree}} = (1-\alpha)\varepsilon_{\text{total}}. \tag{17}$$

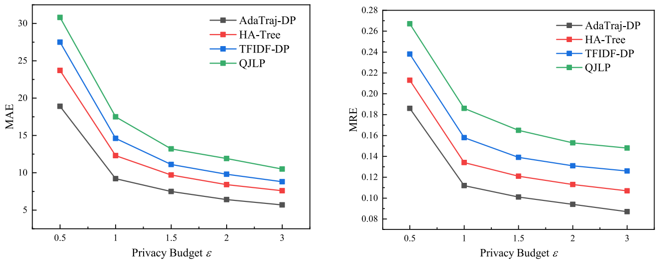
We vary $\varepsilon_{\text{total}}$ from 0.5 to 3.0 to investigate the privacy–utility trade-off.

The variance factor γ controlling dynamic budget adaptation is selected from $\{0, 0.2, 0.5, 1.0\}$, and the hierarchical smoothing parameter is set to $a = 1.0$. The sensitivity threshold θ_S for classifying sensitive points is chosen from $\{0.6, 0.7, 0.8, 0.9\}$. The personalized budget range is fixed at $[\varepsilon_{\min}, \varepsilon_{\max}] = [0.1, 1.0]$.

To ensure comparability, all methods share identical grid resolution ($G = 128$) and Hilbert encoding length ($L_{\text{enc}} = 16$). All experiments are implemented in Python 3.8 with PyTorch 2.4 on an NVIDIA RTX 4090 GPU.

5.2. RQ1: Data utility evaluation

This experiment evaluates how AdaTraj-DP preserves the analytical utility of published trajectories under different privacy budgets. All



(a) MAE of Count Queries

(b) MRE of Count Queries

Fig. 2. Trajectory count query accuracy under varying $\varepsilon_{\text{total}}$ on both datasets.

evaluations are conducted on both the T-Drive and GeoLife datasets, covering dense and sparse mobility scenarios to ensure cross-domain consistency.

5.2.1. Accuracy of trajectory count queries

We evaluate the ability of each method to answer prefix-based count queries accurately. For each dataset, a query set Q consisting of 1000 random trajectory prefixes with lengths between 4 and 8 is selected. Let $c(q)$ denote the true count of trajectories matching prefix $q \in Q$, and $\hat{c}(q)$ be the noisy count returned by the mechanism. The data utility is quantified using Mean Absolute Error (MAE) and Mean Relative Error (MRE), defined as:

$$\text{MAE} = \frac{1}{|Q|}\sum_{q\in Q}|c(q)-\hat{c}(q)|,\ \ \text{MRE} = \frac{1}{|Q|}\sum_{q\in Q}\frac{|c(q)-\hat{c}(q)|}{\max(c(q),\delta)} \tag{18}$$

where δ is a smoothing parameter (set to 1% of the total dataset size) to prevent division by zero for small counts. The results are averaged over ten repetitions with independent noise realizations.

Effect of Privacy Budget $\varepsilon_{\text{total}}$. Figs. 2(a) and 2(b) illustrate the quantitative relationship between privacy strength and data utility. All methods exhibit a convex error decay curve as $\varepsilon_{\text{total}}$ increases from 0.5 to 3.0, reflecting the fundamental differential privacy trade-off.

In the strict privacy regime ($\epsilon_{total} \in [0.5, 1.5]$), our method achieves the steepest marginal reduction in MAE, indicating a high return on privacy budget investment. Specifically, when ϵ_{total} increases from 0.5 to 1.0, AdaTraj-DP reduces the MAE by approximately 45.3% (from 18.1 to 9.9), whereas the second-best baseline, HA-Tree, only achieves a 31.4% reduction. This quantitative gap demonstrates that AdaTraj-DP yields a significantly higher marginal utility gain for every unit of privacy budget expended compared to static hierarchical structures.

5.2.2. Preservation of spatial distribution

Spatial fidelity evaluates the geometric similarity between the original and perturbed trajectories. We use two complementary metrics: the *Hausdorff Distance (HD)* for worst-case deviation and the *Mean Displacement (MD)* for average positional distortion.

Effect of Privacy Budget $\varepsilon_{\text{total}}$. Fig. 3 and Table 1 summarize the spatial accuracy across privacy levels. For both T-Drive and GeoLife datasets, AdaTraj-DP consistently achieves smaller deviations, demonstrating its robustness across data densities and spatial patterns. The sensitivity-guided perturbation preserves local consistency, while adaptive budget redistribution reduces distortion in dense urban regions.

Overall, AdaTraj-DP demonstrates consistent spatial and statistical accuracy across both datasets, validating its generalizability to heterogeneous mobility distributions.

5.3. RQ2: Model utility evaluation

This experiment evaluates how the differentially private trajectories generated by AdaTraj-DP retain their utility for AI-based downstream tasks. Two representative learning tasks are considered: (1) trajectory classification, which predicts the semantic category of a movement sequence; (2) destination prediction, which estimates the likely endpoint of an ongoing trajectory. These tasks are evaluated on the T-Drive and GeoLife datasets to reflect both dense and sparse urban mobility environments.

Table 1
Spatial fidelity comparison (average over T-Drive and GeoLife datasets). Lower values indicate higher spatial accuracy.

$\varepsilon_{\text{total}}$	Hausdorff Distance (HD)		Mean Displacement (MD)	
	AdaTraj-DP	Best Baseline	AdaTraj-DP	Best Baseline
0.5	0.152	0.171 (HA-Tree)	0.098	0.113 (HA-Tree)
1.0	0.096	0.127 (HA-Tree)	0.069	0.087 (HA-Tree)
1.5	0.089	0.125 (TFIDF-DP)	0.063	0.088 (TFIDF-DP)
2.0	0.083	0.118 (TFIDF-DP)	0.059	0.083 (TFIDF-DP)
3.0	0.079	0.130 (QJLP)	0.056	0.094 (QJLP)

5.3.1. Trajectory classification

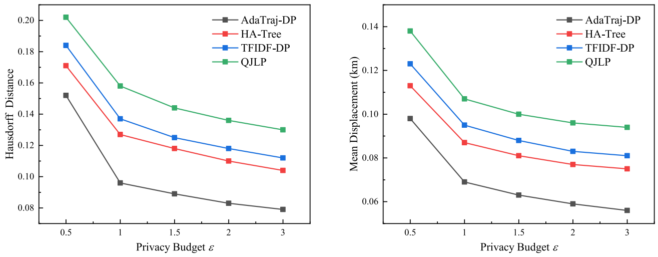
A hierarchical Transformer-based model with positional encoding is trained on the published trajectories to perform multi-class trajectory classification. The model architecture follows a standard encoder setup with three attention layers and a hidden size of 256. Each experiment is repeated five times under independent noise realizations, and the average classification accuracy and macro F1-score are reported. The total privacy budget $\varepsilon_{\text{total}}$ is varied from 0.5 to 3.0.

Effect of Privacy Budget $\varepsilon_{\text{total}}$. Figs. 4(a) and 4(b) illustrate the influence of $\varepsilon_{\text{total}}$ on model performance. As the privacy budget increases, both accuracy and F1-score improve across all methods. AdaTraj-DP consistently maintains the highest model utility on both datasets, demonstrating that adaptive sensitivity control effectively preserves discriminative features. The hierarchical tree representation mitigates local noise accumulation, supporting stable model convergence.

5.3.2. Destination prediction

To evaluate predictive consistency, a sequence-to-sequence neural decoder is trained to predict the destination region of each trajectory prefix. Prediction accuracy is measured by the top-1 hit rate, while spatial accuracy is quantified by the mean geodesic distance between predicted and true destinations.

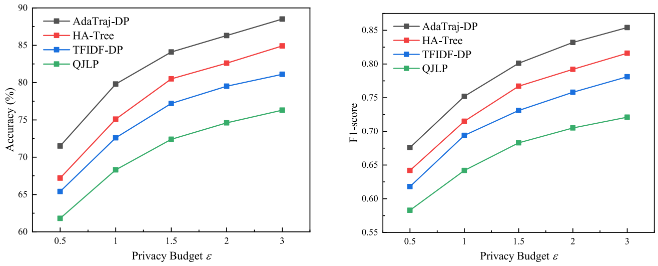
Effect of Privacy Budget $\varepsilon_{\text{total}}$. Figs. 5(a) and 5(b) illustrate the results of destination prediction across both datasets. AdaTraj-DP maintains stable predictive performance even under strict privacy constraints ($\varepsilon_{\text{total}} < 1.0$), consistently outperforming fixed-budget baselines that cannot adapt to local sensitivity variations. As the privacy budget increases, the prediction accuracy steadily improves, while the mean spatial deviation between predicted and true destinations decreases. This demonstrates that adaptive perturbation and hierarchical encoding together preserve mobility semantics and ensure downstream models can effectively capture trajectory intent despite injected noise.



(a) Hausdorff Distance vs. Privacy Budget

(b) Mean Displacement vs. Privacy Budget

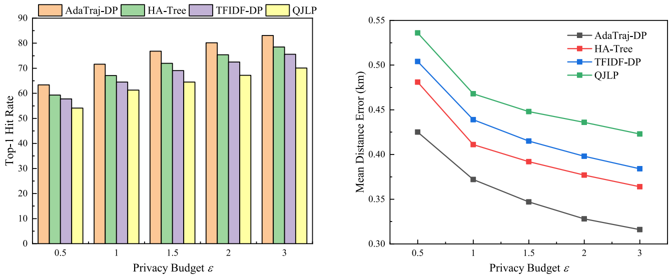
Fig. 3. Spatial fidelity comparison on T-Drive and GeoLife datasets.



(a) Classification Accuracy

(b) F1-score

Fig. 4. Trajectory classification performance under varying $\varepsilon_{\text{total}}$ on T-Drive and GeoLife datasets.



(a) Destination Prediction Accuracy (Top-1 Hit Rate)

(b) Destination Prediction Mean Distance Error (km)

Fig. 5. Destination prediction accuracy and spatial deviation under varying $\varepsilon_{\text{total}}$ on T-Drive and GeoLife datasets.

5.4. RQ3: Parameter sensitivity analysis

This experiment investigates the effect of key parameters in AdaTraj-DP on privacy–utility balance, focusing on two critical hyperparameters: the budget allocation ratio α and the sensitivity threshold θ_{TFIDF}. All experiments are conducted with the total privacy budget $\varepsilon_{\text{total}} = 1.5$ on both the T-Drive and GeoLife datasets.

5.4.1. Effect of budget allocation ratio α

The parameter α controls the distribution of the total privacy budget between the point-level perturbation and the hierarchical tree aggregation phases, where $\varepsilon_{\text{point}} = \alpha\varepsilon_{\text{total}}$ and $\varepsilon_{\text{tree}} = (1-\alpha)\varepsilon_{\text{total}}$. A small α assigns more budget to aggregation, reducing hierarchical noise, whereas a large α increases point-level fidelity at the expense of tree consistency. We vary α from 0.1 to 0.9 and evaluate both data utility and model accuracy.

Figs. 6 presents the effect of α on count query error (MAE) and trajectory classification accuracy. An optimal trade-off is observed near $\alpha = 0.6$, where both the query error and model accuracy achieve near-balanced performance. When $\alpha < 0.4$, excessive noise in point perturbation causes degraded spatial precision, while $\alpha > 0.8$ reduces the reliability of aggregated counts in the prefix tree, highlighting the necessity of coordinated budget allocation.

In practice, the optimal α depends on the specific utility requirements. For applications prioritizing fine-grained point precision (e.g., destination prediction), a larger α (e.g., 0.6–0.7) is recommended to allocate more budget to the perturbation phase. Conversely, for range query tasks relying on aggregate statistics, a smaller α favors the hierarchical tree structure. An empirical strategy for parameter selection involves using a small, non-sensitive validation set to estimate the inflection point of the loss function. A balanced initialization of $\alpha = 0.6$ is recommended as a default setting, which prioritizes neither point-level perturbation nor structural aggregation excessively. To ensure privacy integrity, this validation set is constructed from public historical trajectory data (e.g., open-source T-Drive samples) or a disjoint subset of historical records that does not overlap with the private

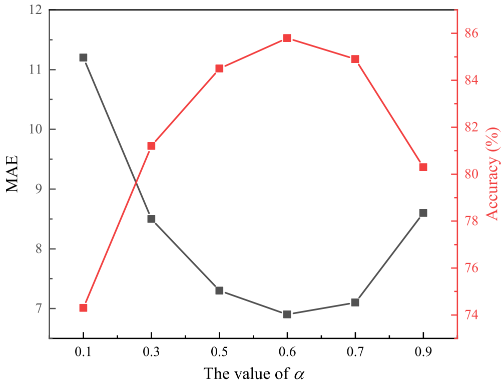


Fig. 6. Impact of budget allocation ratio α on query utility and model performance at $\varepsilon_{\text{total}} = 1.5$.

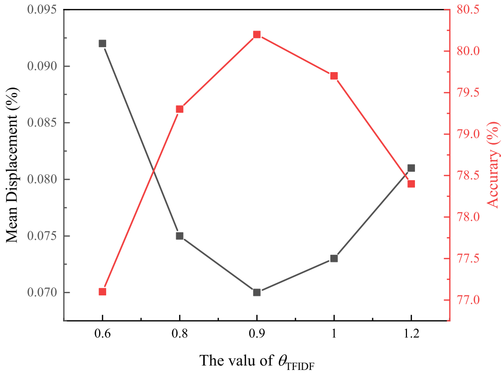


Fig. 7. Effect of the sensitivity threshold θ_{TFIDF} on spatial fidelity and predictive performance at $\varepsilon_{\text{total}} = 1.5$.

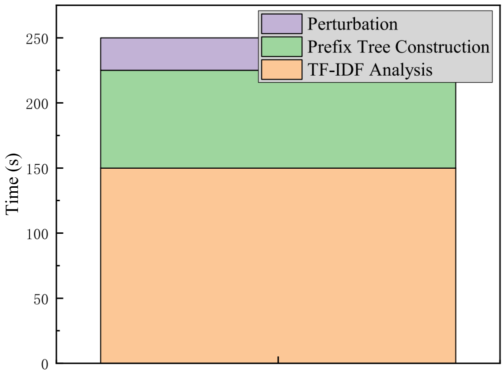


Fig. 8. Computational cost decomposition of AdaTraj-DP across three key stages.

dataset D. This separation guarantees that the hyperparameter tuning process relies solely on public knowledge and does not consume the privacy budget allocated for the sensitive data.

5.4.2. *Effect of sensitivity threshold* θ_{TFIDF}

The threshold θ_{TFIDF} determines how many trajectory points are classified as sensitive during the TF–IDF-based detection process. A smaller threshold labels more points as sensitive, resulting in stronger protection but higher noise magnitude. We vary θ_{TFIDF} from 0.6 to 1.2 and evaluate the mean displacement (MD) and destination prediction accuracy.

Figs. 7 depicts the variation of spatial fidelity and predictive utility under different θ_{TFIDF} values. As θ_{TFIDF} increases, the number of sensitive points decreases, leading to reduced perturbation intensity and smaller average displacement. However, excessively large θ_{TFIDF} weakens privacy coverage and slightly degrades downstream prediction accuracy. The optimal setting is observed around $\theta_{\text{TFIDF}} = 0.9$, balancing spatial accuracy with model generalization.

5.4.3. *Generalization and parameter stability*

In the ablation studies presented above, we observed that the framework's utility is responsive to variations in the budget allocation ratio α and sensitivity threshold θ_{TFIDF}, particularly when these parameters approach the boundaries of their respective ranges. This sensitivity necessitates a discussion on the model's generalization capabilities across different data distributions.

While the framework exhibits sensitivity to extreme parameter variations, it is worth noting that the optimal operating points ($\alpha \approx 0.6$, $\theta_{\text{TFIDF}} \approx 0.9$) remain consistent across both the high-density T-Drive dataset and the sparse, diverse GeoLife dataset. This cross-dataset stability suggests that AdaTraj-DP is robust to heterogeneous spatial distributions, indicating that a standard parameter configuration can yield reliable performance without the need for exhaustive hyperparameter retuning for every new application scenario.

5.5. *Scalability analysis*

To address practical deployment concerns, particularly for city-wide scenarios, we analyze the scalability of AdaTraj-DP regarding both dataset volume (number of users N) and temporal duration (trajectory length L).

Scalability to Large-scale User Datasets. The computational complexity of AdaTraj-DP is dominated by the linear scanning of trajectory points. Specifically, the sensitivity detection and adaptive perturbation phases operate on each trajectory independently, with a time complexity of $O(N \cdot L)$. This independence allows for trivial parallelization across multiple processors, significantly reducing runtime on large-scale datasets. Furthermore, the hierarchical aggregation phase inserts encoded sequences into the prefix tree with a complexity of $O(N \cdot L)$, avoiding the quadratic $O(N^2)$ pairwise comparisons often required by clustering-based or K-anonymity approaches. Consequently, the runtime of AdaTraj-DP grows linearly with the number of users, indicating that the framework is scalable to large-scale spatiotemporal datasets typical of modern urban computing.

Robustness for Long Historical Trajectories. For long historical trajectories, the challenge lies in maintaining structural efficiency and data utility as the sequence length increases. AdaTraj-DP addresses this through two mechanisms:

(1) *Efficient Encoding:* The Hilbert space-filling curve maps high-dimensional spatial points into 1D integers via efficient bitwise operations. Since the encoding complexity is constant per point, the computational cost scales linearly with the trajectory length, avoiding the performance bottlenecks often associated with complex sequence alignment methods.

(2) *Depth-Robust Aggregation:* Long trajectories naturally necessitate deeper prefix trees, which typically suffer from severe budget dilution at lower levels. AdaTraj-DP addresses this through its logarithmic layer-wise allocation (Eq. (12)), which dampens the noise increase rate relative to tree depth. This mechanism ensures that the tail ends of extended mobility sequences retain analytical utility, preventing the rapid signal degradation commonly observed in uniform allocation schemes.

Empirical Efficiency Evaluation. To complement the theoretical complexity analysis, Fig. 8 presents the empirical runtime decomposition of AdaTraj-DP on the T-Drive dataset. The total processing time is approximately 250 s. As observed, the TF–IDF Analysis phase constitutes the majority of the computational overhead (approx. 60%) due to the necessity of global statistical aggregation across the spatial grid. However, the core privacy mechanisms—Prefix Tree Construction and Perturbation—demonstrate high efficiency. Notably, the adaptive perturbation phase accounts for less than 10% of the total time, confirming that the granular noise injection introduces negligible latency. This performance profile validates that AdaTraj-DP is well-suited for periodic batch publishing scenarios (e.g., releasing trajectory updates every 5-10 min for traffic monitoring). While the current execution time is sufficient for such batch-based near-real-time analytics, we acknowledge that strictly latency-critical streaming applications may require further optimization of the tree construction process. Nevertheless, for the targeted high-utility analysis tasks, this computational cost is a justifiable trade-off for the structural consistency provided by the framework.

6. Conclusion

This study presented AdaTraj-DP, an adaptive privacy-preserving framework for publishing trajectory data with differential privacy guarantees. The framework introduces context-aware sensitivity modeling and adaptive budget allocation to balance privacy protection and analytical utility in AI-based mobility analysis. By integrating personalized perturbation with hierarchical prefix-tree aggregation, AdaTraj-DP enables trajectory-level differential privacy while maintaining spatial fidelity and downstream model performance.

Future work will focus on extending AdaTraj-DP to support multimodal trajectory data, integrating semantic and temporal context under unified privacy constraints. Additionally, to address the efficiency concerns in high-frequency streaming environments, we plan to investigate incremental tree update algorithms. This would allow the framework to handle real-time data streams with significantly lower latency while maintaining the established privacy guarantees.

CRediT authorship contribution statement

Yongxin Zhao: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Data curation, Conceptualization. **Chundong Wang:** Writing – review & editing, Project administration, Methodology. **Hao Lin:** Visualization, Validation, Methodology. **Xumeng Wang:** Writing – review & editing, Methodology, Conceptualization. **Yixuan Song:** Methodology, Investigation, Conceptualization. **Qiuyu Du:** Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Thanks to the National Key R&D Program of China (2023YFB2703900).

Appendix. Conversion from integer values to binary sequences

Our prefix tree construction necessitates the representation of each geographic coordinate as a character sequence. Although the Hilbert space-filling curve successfully transforms a two-dimensional coordinate $p'_{i,j}$ into a one-dimensional integer $v_{i,j}$, this numerical value cannot be directly incorporated into a conventional prefix tree structure. Consequently, we implement an additional transformation phase that converts this integer into a binary sequence $s_{i,j}$ with fixed length.

This transformation is controlled by the Hilbert curve's order parameter, designated as k. When applying a Hilbert curve with order k, the two-dimensional space becomes divided into a $(2^k) \times (2^k)$ cellular grid. To guarantee that every coordinate within dataset D receives a distinct Hilbert index assignment, the order parameter must fulfill the condition $k \geq \lceil \log \sqrt{|D|} \rceil$. This configuration assigns each cell, including any coordinate it contains, to a unique integer within the interval $[0, (2^k)^2 - 1]$.

The binary sequence length, denoted L_{enc}, depends on the total count of representable integer values. Representing all $(2^k)^2 = 2^{2k}$ distinct values necessitates a binary sequence of length $L_{\text{enc}} = 2k$. The transformation consists of a direct conversion from integer $v_{i,j}$ to its L_{enc}-bit binary form, applying leading zero-padding when needed to maintain uniform length.

Consider the following illustration: assume a Hilbert curve with order $k = 8$. Under these conditions: The cellular count equals $(2^8)^2 = 65{,}536$. The integer value $v_{i,j}$ resides within the interval $[0, 65535]$. The necessary binary sequence length becomes $L_{\text{enc}} = 2 \times 8 = 16$.

When coordinate $p'_{i,j}$ maps to integer $v_{i,j} = 47593$, its 16-bit binary sequence representation becomes:

$$s_{i,j} = \text{Encode}(47593, 16) = "1011100111101001". \tag{A.1}$$

This sequence $s_{i,j}$ serves as the actual element for navigating and constructing the prefix tree. Individual bits within the sequence determine decisions at corresponding tree levels, establishing a multi-level spatial indexing structure. The selection of parameter k (and consequently L_{enc}) represents a crucial design choice that mediates between spatial granularity and the prefix tree's dimensions and computational overhead.

Data availability

Data will be made available on request.

References

[1] W. Zhang, M. Li, R. Tandon, H. Li, Online location trace privacy: An information theoretic approach, IEEE Trans. Inf. Forensics Secur. 14 (1) (2018) 235–250.

[2] F. Jin, W. Hua, M. Francia, P. Chao, M.E. Orlowska, X. Zhou, A survey and experimental study on privacy-preserving trajectory data publishing, IEEE Trans. Knowl. Data Eng. 35 (6) (2022) 5577–5596.

[3] J. Liu, J. Chen, R. Law, S. Wang, L. Yang, Travel patterns and spatial structure: understanding winter tourism by trajectory data mining, Asia Pac. J. Tour. Res. 29 (11) (2024) 1351–1368.

[4] Z. Wu, X. Wang, Z. Huang, T. Zhang, M. Zhu, X. Huang, M. Xu, W. Chen, A utility-aware privacy-preserving method for trajectory publication, IEEE Trans. Vis. Comput. Graphics.

[5] S. Schestakov, S. Gottschalk, T. Funke, E. Demidova, RE-Trace: Re-identification of modified GPS trajectories, ACM Trans. Spat. Algorithms Syst. 10 (4) (2024) 1–28.

[6] C. Dwork, Differential privacy, in: International Colloquium on Automata, Languages, and Programming, Springer, 2006, pp. 1–12.

[7] Z. Yang, R. Wang, D. Wu, H. Wang, H. Song, X. Ma, Local trajectory privacy protection in 5G enabled industrial intelligent logistics, IEEE Trans. Ind. Inform. 18 (4) (2021) 2868–2876.

[8] Z. Shen, Y. Zhang, H. Wang, P. Liu, K. Liu, Y. Shen, BiGRU-DP: Improved differential privacy protection method for trajectory data publishing, Expert Syst. Appl. 252 (2024) 124264.

[9] Y. Zhao, C. Wang, Protecting privacy and enhancing utility: A novel approach for personalized trajectory data publishing using noisy prefix tree, Comput. Secur. 144 (2024) 103922.

[10] S. Yuan, D. Pi, X. Zhao, M. Xu, Differential privacy trajectory data protection scheme based on R-tree, Expert Syst. Appl. 182 (2021) 115215.

[11] W. Cheng, R. Wen, H. Huang, W. Miao, C. Wang, OPTDP: Towards optimal personalized trajectory differential privacy for trajectory data publishing, Neurocomputing 472 (2022) 201–211.

[12] N. Niknami, M. Abadi, F. Deldar, A fully spatial personalized differentially private mechanism to provide non-uniform privacy guarantees for spatial databases, Inf. Syst. 92 (2020) 101526.

[13] P. Liu, D. Wu, Z. Shen, H. Wang, K. Liu, Personalized trajectory privacy data publishing scheme based on differential privacy, Internet Things 25 (2024) 101074.

[14] W. Qardaji, W. Yang, N. Li, Differentially private grids for geospatial data, in: 2013 IEEE 29th International Conference on Data Engineering, ICDE, IEEE, 2013, pp. 757–768.

[15] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, T. Yu, Differentially private spatial decompositions, in: 2012 IEEE 28th International Conference on Data Engineering, IEEE, 2012, pp. 20–31.

[16] J. Hua, Y. Gao, S. Zhong, Differentially private publication of general time-serial trajectory data, in: 2015 IEEE Conference on Computer Communications, INFOCOM, IEEE, 2015, pp. 549–557.

[17] Z. Zhang, X. Xu, F. Xiao, LGAN-DP: A novel differential private publication mechanism of trajectory data, Future Gener. Comput. Syst. 141 (2023) 692–703.

[18] Y. Hu, Y. Du, Z. Zhang, Z. Fang, L. Chen, K. Zheng, Y. Gao, Real-time trajectory synthesis with local differential privacy, in: 2024 IEEE 40th International Conference on Data Engineering, ICDE, IEEE, 2024, pp. 1685–1698.

[19] R. Zhang, W. Ni, N. Fu, L. Hou, D. Zhang, Y. Zhang, DP-LTGAN: Differentially private trajectory publishing via Locally-aware Transformer-based GAN, Future Gener. Comput. Syst. 166 (2025) 107686.

[20] S. Jiao, J. Cheng, Z. Huang, T. Li, T. Xie, W. Chen, Y. Ma, X. Wang, DPKnob: A visual analysis approach to risk-aware formulation of differential privacy schemes for data query scenarios, Vis. Inform. 8 (3) (2024) 42–52.

[21] X. Wang, S. Jiao, C. Bryan, Defogger: A visual analysis approach for data exploration of sensitive data protected by differential privacy, IEEE Trans. Vis. Comput. Graphics 31 (1) (2025) 448–458, http://dx.doi.org/10.1109/TVCG.2024.3456304.

[22] R. Chen, B.C.M. Fung, B.C. Desai, Differentially private trajectory data publication, 2011, arXiv:1112.2020, URL https://arxiv.org/abs/1112.2020.

[23] C. Yin, J. Xi, R. Sun, J. Wang, Location privacy protection based on differential privacy strategy for big data in industrial internet of things, IEEE Trans. Ind. Inform. 14 (8) (2017) 3628–3636.

[24] Y. Zhao, C. Wang, E. Zhao, X. Zheng, H. Lin, PerTrajTree-DP: A personalized privacy-preserving trajectory publishing framework for trustworthy AI systems, in: Data Security and Privacy Protection, Springer Nature Singapore, Singapore, ISBN: 978-981-95-3182-0, 2026, pp. 57–75.

[25] T. Wang, Y. Tao, A. Gilad, A. Machanavajjhala, S. Roy, Explaining differentially private query results with dpxplain, Proc. VLDB Endow. 16 (12) (2023) 3962–3965.

[26] Z. Huang, J. Liu, D.G. Alabi, R.C. Fernandez, E. Wu, Saibot: A differentially private data search platform, Proc. VLDB Endow. (PVLDB) 16 (11) (2023) PVLDB 2023 demo / system paper.

[27] Y. Dai, J. Shao, C. Wei, D. Zhang, H.T. Shen, Personalized semantic trajectory privacy preservation through trajectory reconstruction, World Wide Web 21 (2018) 875–914.

[28] K. Zuo, R. Liu, J. Zhao, Z. Shen, F. Chen, Method for the protection of spatiotemporal correlation location privacy with semantic information, J. Xidian Univ. 49 (1) (2022) 67–77.

[29] S. Denisov, H.B. McMahan, J. Rush, A. Smith, A. Guha Thakurta, Improved differential privacy for sgd via optimal private linear operators on adaptive streams, Adv. Neural Inf. Process. Syst. 35 (2022) 5910–5924.

[30] H. Fang, X. Li, C. Fan, P. Li, Improved convergence of differential private sgd with gradient clipping, in: The Eleventh International Conference on Learning Representations, 2023.

[31] J. Fu, coauthors, DPSUR: Accelerating differentially private training via selective updates and release, Proc. VLDB Endow. (PVLDB) 17 (2024) PVLDB paper; PDF available from VLDB site.

[32] Y. Zheng, Trajectory data mining: an overview, ACM Trans. Intell. Syst. Technol. (TIST) 6 (3) (2015) 1–41.

[33] M.E. Andrés, N.E. Bordenabe, K. Chatzikokolakis, C. Palamidessi, Geo-indistinguishability: Differential privacy for location-based systems, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, 2013, pp. 901–914.

[34] W. Zhang, M. Li, R. Tandon, H. Li, Semantic-aware privacy-preserving online location trajectory data sharing, IEEE Trans. Inf. Forensics Secur. 17 (2022) 2292–2306.

[35] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, Y. Huang, T-drive: driving directions based on taxi trajectories, in: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2010, pp. 99–108.

[36] Y. Zheng, X. Xie, W.-Y. Ma, et al., GeoLife: A collaborative social networking service among user, location and trajectory, IEEE Data Eng. Bull. 33 (2) (2010) 32–39.

[37] Y. Zhao, C. Wang, L. Li, X. Wang, H. Lin, Z. Liu, TrajMamba: A multi-scale mamba-based framework for joint trajectory and road network representation learning, 2025, https://ssrn.com/abstract=5624451.