Revisit Location Privacy Protection: A Service Failure Perspective

Shaowei Wang
wangsw@gzhu.edu.cn
Guangzhou University
Guangzhou, Guangdong, China

Han Wu
h.wu@soton.ac.uk
University of Southampton
Southampton, UK

Yun Peng
yunpeng@gzhu.edu.cn
Guangzhou University
Guangzhou, Guangdong, China

Puning Zhao
zhaopn@mail.sysu.edu.cn
Sun Yat-sen University (Shenzhen)
Shenzhen, Guangdong, China

Wenqi Ren
renwq3@mail.sysu.edu.cn
Sun Yat-sen University (Shenzhen)
Shenzhen, Guangdong, China

Changyu Dong
changyu.dong@gzhu.edu.cn
Guangzhou University
Guangzhou, Guangdong, China

Jin Li
jinli71@gmail.com
Guangzhou University
Guangzhou, Guangdong, China

Jian Weng
cryptjweng@gmail.com
Guangzhou University
Guangzhou, Guangdong, China

Abstract

Protecting location privacy is essential in many real-world applications and has been widely investigated. However, existing solutions face a fundamental trade-off. Heuristic techniques like location anonymization, dummy generation, obfuscation, or generalization lack rigorous privacy guarantees and remain vulnerable to attacks. In contrast, formal approaches like local differential privacy and geo-indistinguishability introduce service failures (i.e. the distance between the reported and true location exceeds a threshold r). The probability of such failures can be significant (e.g., $\geq 20\%$) under practical privacy budgets. This tension leads to a central question:

Can we formally protect location privacy without incurring failures?

We answer this question affirmatively through PLODA, a novel framework that protects user location privacy via three key mechanisms: (i) bounded-range location obfuscation, (ii) dummy injection, and (iii) report anonymization. Our framework achieves differential privacy while keeping the obfuscation ranges small enough to ensure zero failure probability. It also incurs minimal overhead (e.g., fewer than 0.5 dummies per user in typical scenarios) and maintains high utility. To accommodate heterogeneous user requirements, we further propose a personalized privacy-failure trade-off framework based on user-specific failure radii. Experimental results demonstrate that our approach consistently outperforms existing methods.

CCS Concepts

• **Security and privacy → Privacy-preserving protocols**.

Keywords

differential privacy, local differential privacy, location data, LBS

ACM Reference Format:
Shaowei Wang, Han Wu, Yun Peng, Puning Zhao, Wenqi Ren, Changyu Dong, Jin Li, and Jian Weng. 2018. Revisit Location Privacy Protection: A Service Failure Perspective. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 24 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Location-based services (LBS) are pervasive across numerous application domains, including navigation systems, location-aware advertising, social networks, and virtual/augmented reality. These services fundamentally rely on user location data to deliver context-aware functionality. Reporting precise user locations, however, introduces significant privacy risks, such as movement pattern reconstruction, identity inference attacks, and point-of-interest exposure (e.g., medical facilities or personal residences).

Existing works. Classical location-privacy techniques include anonymization [26, 29], dummy queries[40, 57], spatial obfuscation [4, 5], and generalization [27, 46]. Specifically, anonymization obscures user identity; dummy queries inject fake locations; obfuscation replaces exact coordinates with imprecise locations; and generalization reduces location granularity. These approaches are widely deployed in real-world systems (e.g., Google [32], Apple [31], and Huawei [33]). For example, Apple Maps [31] converts locations to less precise ones at the start of GPS navigation, while routing and traffic data are processed in an anonymized and aggregated form. Similarly, Android OS allows users to control location accuracy, and Google uses anonymized locations for Ads service [32]. However, these methods remain vulnerable to attacks. For instance, studies on Apple [9, 43] and Samsung [60] infrastructures reveal risks of covert tracking, cross-device correlation, and trajectory inference despite anonymization. Likewise, *privacy zones* in Strava [17] can be de-anonymized via multi-activity endpoint inference.

To formally guaranty location privacy, *differential privacy* (DP) and its variants have emerged as a standard, offering resilience against strong adversaries with background knowledge. Aligned with the decentralized architecture of LBS, the *local model* of DP (LDP [39]) is frequently adopted with each user sanitizes their own location data, e.g., via randomized response mechanisms [10, 41, 50]. To optimize privacy-utility tradeoffs, *geo-indistinguishability* (GI, [2]) relaxes LDP indistinguishability constraints for geographically distant locations. This relaxation enables higher distinguishability for distant locations while maintaining strong protection for proximate ones. Geo-indistinguishability is typically achieved by adding planar Laplace noise to location coordinates [2].

The service failure issue. Despite their formal privacy guaranties, LDP-based methods introduce a critical issue in LBS: *service failure*, defined as cases where the distance between the reported and the true location exceeds a threshold r_*. Such failures can incur severe consequences, e.g., incorrect lane guidance in vehicles, posing a substantial risk for critical applications. To satisfy the privacy constraints, LDP mechanisms must assign non-zero probability to reports across the entire domain; consequently, with non-negligible probability, a reported location deviates significantly from the true location. Services relying on accurate positioning (e.g., GPS navigation, proximity-based recommendations) may thus return unusable responses. For instance, even under a rather loose privacy level ($\epsilon = 5$/km) in the relaxed LDP notion ϵ-GI with planar Laplace noise, the reported location deviates from the true location by more than $r_* = 0.5$ km with probability > 0.28.

A naive remedy, repeatedly querying with the DP location mechanism until a "usable" report is generated, breaches privacy: the server infers that the *final* report is proximate to the true location (while prior reports are distant), effectively nullifying the DP guarantee. We note that such repeated reports violate the sequential composition theorem [19] of DP, which assumes *predetermined* number of sequential queries; here, termination depends adaptively on the user's private knowledge of their true location. Another straight-forward approach is posing predetermined number of requests each consumes partial DP budget. This might slightly reduce the failure chance. However, as we theoretically demonstrated in Theorem 3.3, the reduction is limited.

The privacy-failure dilemma. We summarize a fundamental *privacy-failure dilemma* in current location privacy preservation paradigms: (i) Some heuristic methods (anonymization and dummy queries) can avoid service failures by design but are vulnerable to inference attacks; (ii) Existing DP-based approaches provide formal privacy guarantees but induce service failures with non-negligible probability (e.g., assuming a location domain $[-1, 1]^2$, every ϵ-LDP method must incur $\Omega(1/(r_*^2 e^\epsilon))$ failure probability and every ϵ-GI method must incur $\Omega(1 - 1/(\sum_{j\in[0:\lfloor 1/r_*\rfloor-1]}(j+1)\cdot e^{-2r_* j\epsilon}))$ failure probability, with r_* being the failure radius, see Table 1). This dichotomy raises a critical question:

Can we formally protect location privacy without incurring failures?

We answer this question affirmatively, showing that the DP/failure trade-off can be circumvented. Specifically, we propose Private Location Obfuscated Dummy Anonymization (Ploda), integrating:

(1) ***O**bfuscation* with a bounded radius r_*,
(2) ***D**ummy messages* over the location domain,

Table 1: List of location privacy protection methods. The r denotes the failure radius, ϵ is the privacy budget in the local DP and Geo-indistinguishability, (ϵ, δ) is the privacy paramters in the (single-message) PIC shuffle model. Detailed descriptions can be found in Section 3.3.

Method	Differential privacy	Minimax failure probability
anonymization[29]	✗	0
dummy [40, 57]	✗	0
obfuscation[4, 5]	-	-
generalization[46]	-	-
local ϵ-DP[39]	✓	$\geq \frac{\lfloor (1+r)/r\rfloor^2-1}{e^\epsilon+\lfloor (1+r)/r\rfloor^2-1}$
ϵ-GI[2]	✓	$\geq 1-\frac{1}{\sum_{(a,b)\in[0,\lfloor 1/r\rfloor-1]^2} e^{-2r\epsilon\sqrt{a^2+b^2}}}$
PIC shuffle DP[51]	✓	$\geq \frac{(1-\delta)\cdot(\lfloor (1+r)/r\rfloor^2-1)}{n+e^\epsilon\cdot(\lfloor (1+r)/r\rfloor^2-1)}$
Ploda (ours)	✓	**0**

✓: satisfies property ✗: does not satisfy - : indeterminated

(3) ***A**nonymization* of all user messages.

Ploda eliminates service failures by perturbing true locations within radius r_*, ensuring usability. However, bounded perturbation alone permits inference: if a user at $(0.8, 0.8)$ reports within the obfuscation area $[0.6, 1.0]^2$ with $r_* = 0.2$, while others cluster in $[-0.5, 0.5] \times [-0.5, 0.5]$, an adversary can isolate the outlier and accurately infer the location within 0.2. This issue is mitigated by dummy messages and anonymization.

The dummy messages mask possible spatial outliers by ensuring all regions contain plausible reports, at the cost of increased communication overhead. However, this additional overhead remains small under our design. Formally, we prove (ϵ, δ)-DP through a novel analysis based on privacy amplification via shuffling/anonymization [20] tailored for our protocol. Our analysis shows $\tilde{O}\big(\frac{1}{n\epsilon^2 r^2}\big)$ dummies *per user* suffice when user count n is sufficiently large. For typical parameters ($r = 0.15$, $n = 5000$, $\epsilon = 1$, $\delta = 10^{-6}$), this requires < 0.7 dummies per user in expectation. For implementing the anonymization procedure, one might follow real-world anonymous architectures (e.g., by Google [3, 8, 32] and Apple[48]), or use a shuffler layer on the edge side to buffer and permute messages before forwarding [15]. In practice, it fits large-scale deployment, where the number of users n can typically be accumulated over short time windows (e.g., hundreds of users within seconds [6]) to form sufficiently large shuffled batches for effective privacy amplification.

Extensions. Ploda also enables accurate location distribution estimation. This estimation is equally valuable for many location-based services, such as providing crowdsourced statistics (e.g., traffic loads in Apple Maps [31] and Google Maps [32]), gaining insight into service usage and placing resources based on user location patterns (e.g., in Google Ads [32]). Furthermore, considering diverse requirements of users, we support *personalized* failure radii/privacy tradeoffs (likewise the Google Location Accuracy configuration [32] in Android OS): users are allowed to tradeoff their own obfuscating radii r_i and privacy parameters.

Our main contributions are:

- *Failure lower bounds of existing paradigms:* We provide formal lower bound analyses on the failure probability of existing DP approaches. For all possible LDP mechanisms, we show over 89% failure probability in typical settings with $\epsilon = 1$; even with privacy amplification via shuffling [51], and the budget is amplified to $\epsilon = 4$, the failure probability $\geq 30\%$ in typical scenarios. For all GI mechanisms, they failed with $\geq 10\%$ in typical settings.
- *A failure-free DP framework:* We propose Ploda, the first framework for differentially private LBS that eliminates service failures. By novelly integrating classical techniques: obfuscating, dummy, and anonymization, in a carefully sequenced pipeline, we achieve formal (ϵ, δ)-DP guarantees and zero failure probability. We also present nearly tight privacy analyses, demonstrating only $\tilde{O}(1/n)$ dummy message complexity per user.
- *Accurate distribution estimation:* We show that Ploda can also provide distribution estimation for users' locations, with accuracy nearly matching that of the curator model of differential privacy (and thus is nearly optimal).
- *Flexible failure/privacy criteria:* Within Ploda, we allow users to configure the trade-off between the obfuscation radius and privacy.
- *Empirical validation:* Using real-world datasets, we demonstrate Ploda's excellent utility in both retrieval tasks (location-based retrieval) and statistical tasks (distribution estimation).

Roadmap. Section 2 provides DP preliminaries. Section 3 formalizes LBS requirements and failure definition. Section 4 details the Ploda framework. Section 5 extends the design to personalized privacy-failure tradeoffs. Section 6 presents experimental evaluation. Section 7 retrospects existing techniques. Section 8 discusses limitations and implications. Section 9 concludes.

2 Preliminaries

2.1 Privacy Definitions

Definition 2.1 (Hockey-Stick Divergence [44]). *Let P and Q be two probability distributions on a measurable space $(\mathcal{Z}, \Sigma)$. For $\epsilon \in \mathbb{R}$, the* Hockey-stick divergence *from P to Q is defined as*

$$D_\epsilon(P\|Q) = \int_{z\in\mathcal{Z}} \max\{0, P(z) - e^\epsilon Q(z)\}\, \mathrm{d}z.$$

Differential privacy (DP) ensures that small changes in the input result in limited changes in the output distribution. In the centralized (curator) model, a trusted aggregator collects user records $x_i \in \mathbb{X}$ into a dataset $X = \{x_1, \ldots, x_n\}$ and computes $\mathcal{R}(X)$ using a randomized algorithm $\mathcal{R}$. Two datasets $X, X' \in \mathbb{X}^n$ that differ in one entry are referred to as *neighboring datasets*. Differential privacy requires that the divergence between $\mathcal{R}(X)$ and $\mathcal{R}(X')$ remains below a prescribed threshold (typically, $\delta < O(1/n)$).

Definition 2.2 ((ϵ, δ)-DP [19]). *A randomized mechanism $\mathcal{R} : \mathbb{X}^n \to \mathbb{Y}$ satisfies (ϵ, δ)-DP if, for every pair of neighboring datasets $X, X' \in \mathbb{X}^n$, the following holds: $D_\epsilon\big(\mathcal{R}(X) \,\|\, \mathcal{R}(X')\big) \leq \delta$.*

The local DP (LDP) model requires that each user independently randomizes their own data. More formally, a mechanism $\mathcal{R} : \mathbb{X} \to \mathbb{Y}$ satisfies ϵ-LDP if, for every pair $x, x' \in \mathbb{X}$, the divergence between $\mathcal{R}(x)$ and $\mathcal{R}(x')$ is zero.

Table 2: List of notations.

Notation	Description
$[i]$	$\{1, 2, ..., i\}$
$[i : j]$	$\{i, i+1, ..., j\}$
n	the number of users
m	the number of dummy messages per user
x_i	the data of user i
$\mathbb{X}$	the domain of input datum
$\mathbb{Y}$	the domain of a single message
$\mathbb{B}_r(x)$	the domain of $\{y \mid y \in \mathbb{R}^2 \text{ and } \|y - x\|_2 < r\}$
$\mathbb{A}$	the domain of server-side response
ϵ	the privacy budget
δ	the violating probability of pure DP
$\mathcal{S}$	the shuffling/anonymization procedure
C	the corrupted parties
n^*	the number of honest users
$\mathcal{R}_i$	the input-dependent randomizer of user i
$\mathcal{R}_*$	the dummy generator
$\mathcal{A}$	the server-side algorithm

Definition 2.3 (ϵ-LDP [39]). *A local randomizer $\mathcal{R} : \mathbb{X} \to \mathbb{Y}$ is said to satisfy ϵ-LDP if, for all $x, x' \in \mathbb{X}$: $D_\epsilon\big(\mathcal{R}(x) \,\|\, \mathcal{R}(x')\big) = 0$.*

Geo-indistinguishability (GI) relaxes LDP to protect location data with distance-aware privacy guarantees. For location data, the distance metric $d_{\mathbb{X}}$ can be ℓ_2 distance, ℓ_∞ distance, etc.

Definition 2.4 (ϵ-GI [2]). *Given a distance metric $d_{\mathbb{X}} : \mathbb{X} \times \mathbb{X} \mapsto \mathbb{R}^{\geq 0}$ for a domain $\mathbb{X}$, a local randomizer $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ is said to satisfy ϵ-GI if, for all $x, x' \in \mathbb{X}$: $D_{\epsilon \cdot d_{\mathbb{X}}(x,x')}\big(\mathcal{R}(x) \,\|\, \mathcal{R}(x')\big) = 0$.*

2.2 The Shuffle Model

In the DP context, a useful tool to incorporate anonymization is the shuffle model. Following the randomize-then-shuffle paradigm [7, 11], we model a (multi-message) shuffle-based protocol as $\mathcal{P} = (\mathcal{R}, \mathcal{A})$ where:

- $\mathcal{R} : \mathbb{X} \to \mathbb{Y}^*$ is the local randomizer executed by each client, and
- $\mathcal{A} : \mathbb{Y}^* \to \mathbb{A}$ is the server-side algorithm.

In this framework, $\mathbb{Y}$ denotes the message space, while $\mathbb{A}$ represents the server-side output/response space. During an execution, each user i holds a datum x_i and computes a multiset $Y_i = \mathcal{R}(x_i) \in \mathbb{Y}^*$. The union $Y_1 \cup \ldots \cup Y_n$ is then uniformly permuted by a shuffler $\mathcal{S} : \mathbb{Y}^* \to \mathbb{Y}^*$ before being presented to the analyzer. Thus, the overall protocol may be expressed as:

$$\mathcal{P}(x_1, \ldots, x_n) = \mathcal{A}\Big(\mathcal{S}\big(Y_1 \cup \ldots \cup Y_n\big)\Big).$$

Post-computation communications. Depending on the application setting, the server-side algorithm may be an aggregation function (e.g., for statistical estimation [7, 28]) or may be augmented with an additional communication procedure that returns information to each message sender. Such a procedure may take the form of a duplex communication channel (as in the anonymous architecture of Google Ads [32]) or a non-interactive bulletin board (as in the PIC shuffle model [51]). In location-based information retrieval, the server-side algorithm $\mathcal{A}$ is coupled with a per-message communication procedure, whereas in location distribution estimation, $\mathcal{A}$ is often simply an aggregation function.

Privacy amplification. An appealing feature of the shuffle model is the *privacy amplification* phenomenon. Since the server-side algorithm can be viewed as a post-processing function applied to the shuffled multiset of messages $(Y_1 \cup \cdots \cup Y_n)$, the differential privacy loss of the overall protocol $\mathcal{P}$ is upper bounded by the privacy loss of the shuffled messages themselves. A classical approach for analyzing privacy amplification of shuffled ϵ-LDP reports is the privacy "blanket" framework [7], which identifies a common component in the message distributions of non-target users that acts as a "blanket" to help conceal a victim user's data. Recent work [21] shows that when each user applies the same ϵ-LDP mechanism, shuffling n such messages yields an amplified privacy guarantee of $(O(\sqrt{e^\epsilon \log(1/\delta)/n}), \delta)$-DP. In this work, however, we instead employ specialized local randomizers that have bounded single-report error but do not satisfy ϵ-LDP, which makes the privacy analysis more involved. In particular, we must explicitly derive the resulting privacy guarantee using techniques analogous to those in the privacy "blanket" framework.

3 Problem Formulation

3.1 Motivating Applications

We describe two prevalent exemplar computation tasks in LBS: location-based information retrieval and location distribution estimation. Each task involves three key steps:

Location-based retrieval services. A location-based information retrieval system, such as map services or finding nearby restaurants, typically involves the following steps:

I. *Query submission:* Each user $i \in G_a$ submits a query x_i, where x_i denotes the user's location.
II. *Data processing:* The server processes the set of queries $\{x_i\}_{i \in G_a}$ and retrieves results (e.g., nearby restaurants, shortest paths).
III. *Result delivery:* Each user retrieves the query results and utilizes the retrieved information (e.g., navigation routes, recommendations, or nearby users).

Location distribution estimation. The systems aggregates spatial data from users to estimate activity intensity (e.g., check-ins data or traffic statistics). The process is as follows:

I. *Data submission:* Each user $i \in G_a$ submits location x_i.
II. *Data aggregation:* The server aggregates the spatial submissions $\{x_i\}_{i \in G_a}$ and generates a distribution estimate.
III. *Operation:* The server uses the estimated location distribution to take operational actions (e.g., reallocating service resources).

In many scenarios, two tasks can occur simultaneously. For instance, in navigation map [31] or mobile Ads [32] service, the server offers location-based information retrieval services, meanwhile collecting location statistics to analyze the geographical usage distribution of its services and to improve the quality of services.

3.2 Failures in Privacy-preserving LBS

Privacy protection injects noise into location reports, causing spatial distortion that may lead to LBS functional failures, especially in applications requiring high precision. For example, strong privacy guarantees in navigation can cause large deviations, potentially diverting ambulances in emergencies and causing life-threatening delays. Even routine nearby restaurant searches can yield unreachable recommendations, degrading user experience and trust.

Following the failure definition in the seminal work on geo-indistinguishability [2] (which assumes a single query per user), we formalize location service failure as the event that all queries from a user are at least distance r_* (termed as *failure radii*) from the true location x:

Definition 3.1 (Service Failure). *A* service failure *occurs for user i when all location reports submitted by the user deviate by at least distance r_* from their true location. Formally, let x_i is the true location of user i, for the set of reports Y_i from user i, if*

$$\min_{y \in Y_i} \|y - x_i\|_2 \geq r_*,$$

then the service has failed for that user.

We note that, for information retrieval tasks, a service is considered successful as long as at least one report falls within the acceptable radius r_*. Hence, a service failure occurs only when all reports lie outside the r_*-radius neighborhood of the true location.

Location distribution estimation is also affected by the out of r_*-radii noise. Privacy protection methods such as LDP [2, 50] map each true location to a random output across the entire domain. Although debiasing can be applied to distribution estimation using the mapping probability matrix (e.g., in [10, 50]), the variance introduced by large norms in the inverse of this matrix is substantial [53]. Actually, LDP inherently exhibits a significant utility gap compared with curator DP when estimating distributions [18]. Excessive noise in estimated distributions can obscuring genuine high-density regions (e.g., rush-hour subway stations), and mislead downstream decision-making (e.g., public transit dispatching).

3.3 Failure Lower Bounds of Existing Methods

Service failures in privacy-preserving LBS stem from a fundamental tension in existing DP approaches. DP requires assigning nonzero probability across the entire output space. This requirement conflicts with the bounded precision constraints inherent to LBS.

Formally, consider a two-dimension location domain $[-1, 1]^2$, and a r-radii failure definition for each input. Then, for any ϵ-LDP randomizer, the minimax failure probability is at least $1/(r^2 e^\epsilon + 1)$ (see Theorem 3.2). For example, when $r = 0.2$ and $\epsilon = 1$, the probability is at least 89%. When $r = 0.2$ and $\epsilon = 5$, the probability is at least 13%, which remains unacceptable in practice. When each user submits multiple requests that each consume part of the LDP budget, the failure probability may be slightly reduced in some cases (see Theorem 3.3). For instance, for $r = 0.2$, $\epsilon = 1$, $k = 5$, and $\epsilon_j = 1/5$, the probability bound falls to 75%; for $r = 0.2$, $k = 5$, and $\epsilon_1 = 1$ and $\epsilon_j = 0$ for $j \in [2:5]$, the probability bound further falls to 73%, which however is still an unacceptable rate.

Theorem 3.2 (Failure lower bound of LDP). *Given $r \in (0, 1]$, then for any ϵ-LDP randomizer $\mathcal{R} : [-1, 1]^2 \mapsto \mathbb{R}^2$, we have:*

$$\max_{l \in [-1,1]^2} \mathbb{P}[\|\mathcal{R}(l) - l\|_2 \geq r] \geq \frac{\lfloor (1+r)/r \rfloor^2 - 1}{e^\epsilon + \lfloor (1+r)/r \rfloor^2 - 1}.$$

Theorem 3.3 (Failure lower bound of LDP with multiple requests). *Given $r \in (0, 1]$, for any k randomizers $\{\mathcal{R}_1, \ldots, \mathcal{R}_k\}$*

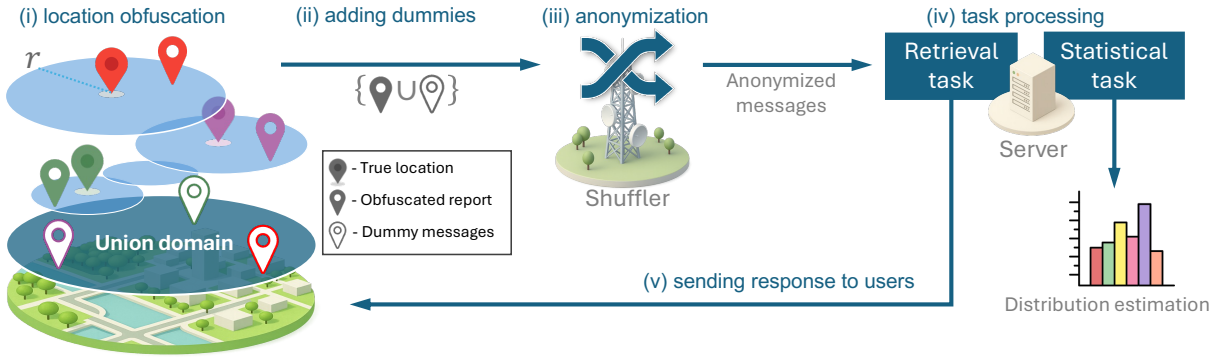

Figure 1: Overview of PLODA framework. Each user first performs bounded obfuscation within an r-radius region (i), then adds input-independent dummy reports sampled from the union domain (ii). The shuffler anonymizes all messages across users (iii) before the server processes them for retrieval or statistical tasks (iv), and return results to users (v).

where $\mathcal{R}_j : [-1,1]^2 \mapsto \mathbb{R}^2$ *satisfies* ϵ_j*-LDP and* $\epsilon = \sum_{j\in[k]} \epsilon_j$*, we have* $\max_{l\in[-1,1]^2} \mathbb{P}[(\min_{j\in[k]} \|\mathcal{R}_j(l) - l\|_2) \geq r]$ *no less than:*

$$1 - \sum_{j\in[k]} \frac{1}{1 + e^{-\epsilon_j} \cdot (\lfloor (1+r)/r \rfloor^2 - 1)}.$$

A similar phenomenon also arises under ϵ-GI, see Theorem 3.4 for the corresponding lower bounds on the failure probability. For example, for $r = 0.2$ and $\epsilon = 3$, the failure bound is 56%; for $r = 0.2$ and $\epsilon = 5$, the probability bound falls to 28%.

THEOREM 3.4 (FAILURE LOWER BOUND OF GI). *Given* $r \in (0,1]$*, then for any* ϵ*-geo-indistinguishability randomizer* $\mathcal{R} : [-1,1]^2 \mapsto \mathbb{R}^2$ *using the* ℓ_2 *distance metric, the following inequality holds:*

$$\max_{l\in[-1,1]^2} \mathbb{P}[\|\mathcal{R}(l) - l\|_2 \geq r] \geq 1 - \frac{1}{\sum_{(a,b)\in[0:\lfloor 1/r \rfloor - 1]^2} e^{-2r\epsilon\sqrt{a^2+b^2}}}.$$

Even with privacy amplification via shuffling, such as in the PIC model [51] where each user sends an anonymous LDP location, the resulting failure probability lower bound is still non-negligible. In particular, Theorem 3.5 (with proof deferred to Appendix D) shows that the minimax failure lower bound in the PIC shuffle model is $\frac{(1-\delta)\cdot(\lfloor (1+r)/r \rfloor^2 - 1)}{n + e^{\epsilon}\cdot(\lfloor (1+r)/r \rfloor^2 - 1)}$. For example, with $n = 100$, $r = 0.2$, $\epsilon = 1$, and $\delta = 0.01/n$, the failure probability is at least 17.9%. With $n = 1000$, $r = 0.1$, $\epsilon = 1$, and $\delta = 0.01/n$, the failure probability is at least 9.0%.

THEOREM 3.5 (FAILURE LOWER BOUND IN SINGLE-MESSAGE SHUFFLE MODEL). *Given* $r \in (0,1]$*, then for any randomizer* $\mathcal{R} : [-1,1]^2 \mapsto \mathbb{R}^*$ *such that* $\mathcal{S} \circ \mathcal{R}(X)$ *and* $\mathcal{S} \circ \mathcal{R}(X')$ *are* (ϵ, δ)*-indistinguishable for all possible neighboring datasets* $X, X' \in [-1,1]^{2\times n}$*, and for any estimator* $f : \mathbb{R}^* \mapsto \mathbb{R}^2$*, the following inequality holds:*

$$\max_{l\in[-1,1]^2} \mathbb{P}[\|f(\mathcal{R}(l)) - l\|_2 \geq r] \geq \frac{(1-\delta)\cdot(\lfloor (1+r)/r \rfloor^2 - 1)}{n + e^{\epsilon}\cdot(\lfloor (1+r)/r \rfloor^2 - 1)}.$$

4 Framework and Protocol Design

In this section, we introduce a general framework, *Private Location Obfuscated Dummy Anonymization* (PLODA), for privacy-preserving LBS. We then present a concrete protocol instantiation and provide a theoretical analysis of its privacy and utility guarantees.

4.1 A General Framework

As illustrated in Fig. 1, the PLODA framework consists of five major steps: obfuscating user data, adding dummies, anonymization, server-side task processing, and sending responses (for retrieval tasks). Specifically, the obfuscating and dummy-adding procedure are performed on the user side, while anonymization is handled by the shuffler. The shuffler can be implemented via anonymous communication channels, cellular stations, edge servers, or cryptographic shuffle servers (see [15] for an excellent survey). The anonymized messages are then processed by the server.

Step (i): Location obfuscation. Each honest user i perturbs their location data $x_i \in \mathbb{R}^2$ to a randomized value y_i near the true input. For instance, $y_i \in \mathbb{Y}_i$ can be sampled uniformly at random from the r-radius neighborhood $\{y \mid y \in \mathbb{R}^d \text{ and } \|y - x_i\|_2 \leq r\}$ around x_i. We refer to y_i as the *input-dependent message* of user i.

Step (ii): Adding dummies. Each honest user i additionally generates m dummy messages M_i, each is sampled from the output domain $\mathbb{Y}$. The $\mathbb{Y}$ should cover at least all possible values of input-dependent messages, so that real messages remain plausible.

Step (iii): Message anonymization. The shuffler receives all input-dependent and dummy messages from the n users, uniform-randomly permutes them, and forwards them to the server.

Step (iv): Server-side task processing. After receiving the shuffled messages, the server performs task-dependent computation. For retrieval tasks, it computes a response for each message (e.g., nearby road maps or restaurant recommendations) and produces an output for every received message, including dummy ones. For statistical tasks, the server aggregates the messages to obtain summary statistics (e.g., a location distribution).

Step (v): Sending responses. For retrieval tasks, the server sends the computed responses back to users, completing the service workflow. For statistical tasks, the aggregated statistics are typically used for analysis or public reporting rather than returned to individuals.

4.2 Threat Model

We consider two types of attacks: privacy attacks and poisoning attacks. Privacy attacks aim to infer information about victim users, whereas poisoning attacks also seek to manipulate the location

statistics (e.g., poisoning the location distribution to mislead the server's operational decisions).

Security model. We consider an adversary who controls a set of corrupted users $C \subset [n]$ and possesses the protocol's public parameters $\mathbb{X}, \mathbb{Y}$ as well as the true data of users in C. In Steps (i) and (ii), the adversary can send arbitrarily designated message contents to the shuffler, though the total number of messages per user remains limited to $\lceil m \rceil + 1$ (e.g. using anonymous message rate limitation on the shuffler [16]). We assume corrupted messages are restricted to the message domain $\mathbb{Y}$, as out-of-domain messages can be easily detected and removed. We assume a strong threat model in which the adversary can also observe the anonymized (shuffled) messages from all honest users $[n] \setminus C$ before crafting corrupted messages. The security adversary's goal may be either to compromise the privacy of other users in $[n] \setminus C$ or to contaminate the aggregated location statistics.

Privacy model. Following conventions in the shuffle model, we assume that the adversary can observe the *shuffled* messages from all users and all responses from the server, in addition to the information available to the corrupted users (i.e., messages from the corrupted users C).

Remark 4.1 (Shuffler roles and trust). In PLODA, the shuffler performs two tasks: (i) *random permutation* of incoming messages, which breaks the sender-message linkage; and (ii) *response re-routing* for retrieval tasks, which maps the server's per-message responses back to the originating users via the inverse permutation. Task ii goes beyond the standard shuffle-DP shuffler, which is a stateless permutation invoked once and is typically realizable by a mix-net, because re-routing requires the shuffler to retain the permutation state across the request/response round. The PLODA shuffler is therefore *stateful* (e.g., in Apple's anonymous Ad system [31]), a strictly stronger assumption than the standard stateless shuffler, though still weaker than a fully trusted central curator.

Remark 4.2 (On anonymous rate limitation). Anonymous rate limitation is an orthogonal admission-control layer: it caps the number of messages each user may inject per batch, and can be enforced either shuffler-side or by a separate non-colluding party. Such a primitive is well studied, e.g., via blind-signature tokens [16] or zero-knowledge proof [63], and has been deployed in real-world anonymous systems [48]; it ensures that the LBS server does not re-identify senders while enforcing the limit. We note that rate limitation affects neither honest-user privacy, which holds by the post-processing property in our proof even if adversaries send an arbitrary number of corrupted messages (see Appendix E.1 step i), nor retrieval-query utility, which is preserved by the input-dependent randomizer with failure probability 0. It affects the adversary's influence on aggregate statistics: without such a limit, corrupted users could inject arbitrarily many messages and bias these statistics without bound. If no statistical tasks are run in the LBS, rate limitation can be unnecessary.

4.3 A Protocol Design

In this section, we present a concrete DP protocol for location data with bounded report error (thus avoiding service failures) under the PLODA framework. The privacy guarantee of our approach is analyzed within the shuffle model of DP [7]. To ensure bounded reporting error, we carefully design the obfuscating area and calibrate the number of dummy messages m according to the desired privacy parameters.

Let $\mathbb{B}_r(x)$ denote the ball $\mathbb{B}_r(x) = \{y \mid y \in \mathbb{R}^2 \text{ and } \|y - x\|_2 < r\}$. Given user i's true location $x_i \in \mathbb{X} = \mathbb{B}_1(\vec{0})$, the randomizer $\mathcal{R}_i(\cdot)$ generates $m + 1$ independent messages. One message is dependent on x_i (denoted by $\mathcal{R}_i(x_i)$), and the remaining m messages are input-independent dummy reports (denoted by $\mathcal{R}_*(m)$). Intuitively, the collection of dummy messages across all users serves to hide the input-dependent message $\mathcal{R}_i(x_i)$.

Input-dependent randomizer. To avoid the service failure, we restrict the output domain of the input-dependent report $\mathcal{R}_i(x_i)$ to a region $\mathbb{Y}_i(x_i)$, which is often a nearby area around x_i. Without loss of generality, we define it as $\mathbb{Y}_i(x_i) = \mathbb{B}_{r_i}(x_i)$ with some radii r_i, to comply with the radius-based definition of service failure/success. We here use personalized r_i for the ease of supporting personalized privacy/failure requirements (See Section 5 for detail) but also restrict it in a pre-defined range $(0, \hat{r}]$ (that is, $r_i \in (0, \hat{r}]$). The mechanism $\mathcal{R}_i(x_i)$ then induces a probability density function over $\mathbb{Y}_i(x_i)$. Common choices include the uniform distribution, truncated Laplace distribution, and truncated Gaussian distribution. We define a general input-dependent randomizer $\mathcal{R}_i(x_i)$ as follows:

$$\mathbb{P}[\mathcal{R}_i(x_i) = y] = \begin{cases} 0, & \text{if } y \notin \mathbb{Y}_i(x_i), \\ f_i(x_i, y), & \text{otherwise,} \end{cases} \tag{1}$$

where $f_i(x_i, y) \geq 0$ holds for all $x_i, y \in \mathbb{R}^2$, and $\int_{y \in \mathbb{Y}_i(x_i)} f_i(x_i, y)\, \mathrm{d}y = 1$, so that $\mathcal{R}_i(x_i)$ defines a valid distribution supported on $\mathbb{Y}_i(x_i)$. Commonly-used local randomizers might also satisfy *shift property*: the probability function $f_i(x, y)$ is a simple shift of a template function $f_i(\vec{0}, y)$ and

$$f_i(x_i, y) \equiv f_i(\vec{0}, y - x_i)$$

holds for all possible x_i, y.

Input-independent dummies. To ensure that dummy messages are indistinguishable from true (locally obfuscated) location reports, the dummy output domain must cover the support of every possible input-dependent randomizer. Accordingly, we define the dummy domain so that

$$\mathbb{Y} \supseteq \bigcup_{i \in [n],\, x \in \mathbb{X}} \mathbb{Y}_i(x). \tag{2}$$

For example, if each $\mathbb{Y}_i(x_i)$ is the r_i-radii neighborhood around x_i (with $r_i \in (0, \hat{r}]$), then $\mathbb{Y}$ can be $[-1 - \hat{r}, 1 + \hat{r}]^2$. Each dummy message $y \sim \mathcal{R}_*(1)$ is drawn uniformly at random from $\mathbb{Y}$:

$$\mathbb{P}[\mathcal{R}_*(1) = z] = \begin{cases} 0, & \text{if } z \notin \mathbb{Y}; \\ 1/\mathrm{V}(\mathbb{Y}). & otherwise, \end{cases} \tag{3}$$

where $\mathrm{V}(\mathbb{Y})$ denotes the volume measure of the domain $\mathbb{Y}$.

In our protocol, the expected number of dummies per user, denoted by m, is a real number. This is necessary because the required expected number of dummies is $\tilde{O}(1/n)$, which may fall below 1 when the population size n is large (see Proposition 4.4 for details). For real-valued m, the dummy-generation procedure $\mathcal{R}_*(m)$ can be implemented as follows:

(i) sample an integer value $m' \sim Binomial(\lceil m \rceil, \frac{m}{\lceil m \rceil})$,
(ii) generate m' dummies each follows $\mathcal{R}_*(1)$.

Algorithm 1: PLODA Protocol Workflow

Input: m: expected dummy count; $\mathbb{Y}$: output domain; $\mathcal{R}_i$: user-side local randomizers for $i \in [n]$

// User Side

foreach *user* $i \in [n] \backslash C$ **do**
  $y_i \leftarrow \mathcal{R}_i(x_i)$ // Step (i): Bounded obfuscation
  $m_i \sim \text{Binomial}(\lceil m \rceil, \frac{m}{\lceil m \rceil})$
  $M_i \leftarrow \emptyset$
  for $j \leftarrow 1$ **to** m_i **do**
    $z_j \sim \text{Uniform}(\mathbb{Y})$ // Step (ii): Dummy generation
    $M_i \leftarrow M_i \cup \{z_j\}$
  $Z_i \leftarrow \{y_i\} \cup M_i$
  send Z_i to shuffler
foreach *user* $i \in C$ **do**
  // possibly utilized information $\mathcal{S}(\{Z_i\}_{i \in [n] \backslash C})$
  $Z_i \leftarrow$ arbitrary multiset of $\leq \lceil m \rceil + 1$ elements from $\mathbb{Y}$
  send Z_i to shuffler

// Shuffler Side

$Z \leftarrow \bigcup_{i=1}^{n} Z_i$
$Z_{\text{perm}} \leftarrow \text{RandomPermutation}_{\pi}(Z)$ // Step (iii): Anonymization using a random permutation π
Send Z_{perm} to server

// Server Side step (iv)

if *task is information retrieval* **then**
  foreach $y \in Z_{perm}$ **do**
    $r_y \leftarrow \text{ProcessRequest}(y)$ // Retrieval task
  Send $\{r_y\}$ to shuffler
else
  stats $\leftarrow \text{Aggregate}(Z_{\text{perm}})$ // Statistical task

// Shuffler Response (Retrieval Task)

foreach r_y **do**
  $i \leftarrow \pi^{-1}(y)$ // Map to original user
  Send r_y to user i

// User Response (Retrieval Task)

foreach *user* $i \notin C$ **do**
  Extract response of y_i from received responses

Equivalently, we define $\hat{\mathcal{R}}_*(1)$ that can generate a special empty output $\perp$ as follows:

$$\hat{\mathcal{R}}_*(1) = \begin{cases} \text{uniform}(\mathbb{Y}), & \text{with prob. } m/\lceil m \rceil \\ \perp. & otherwise. \end{cases} \tag{4}$$

Let $\hat{\mathcal{R}}_*(k)$ denote k independent invocations of $\hat{\mathcal{R}}_*(1)$, where $k \in \mathbb{N}^+$. Then $\hat{\mathcal{R}}_*(\lceil m \rceil) \overset{d}{=} \mathcal{R}_*(m)$, where $\overset{d}{=}$ denotes equality in distribution after ignoring the $\perp$ symbols.

The overall protocol is summarized in Algorithm 1. On the server side, out-of-domain reports can be projected to the nearest point in the input domain $[-1, 1]^2$.

4.4 Privacy Analyses

In this part, we derive the differential privacy guarantees of randomizers used in the proposed protocol. We first establish a numerical privacy bound for the protocol with general local randomizers (see Section 4.4.1). For commonly-used input-dependent randomizers that additionally satisfy the shift property, this numerical bound can be simplified by only analyzing the template distribution (see Section 4.4.2). Furthermore, when the input-dependent randomizer is uniform over the r_i-radii domain $\mathbb{Y}_i(x_i)$, we provide a more efficient numerical bound in Theorem 4.3, as well as a concise closed-form bound in Proposition 4.4.

4.4.1 Differential Privacy Guarantees for General Randomizers. In the presence of corrupted parties C, there remain $n - |C \cap [n]|$ honest users. For convenience, let n^* denote $n - |C \cap [n]|$. These users contribute n^* input-dependent messages and approximately $n^* \cdot m$ dummy messages. In our setting, differential privacy means that for any pair of neighboring datasets $X = (x_1, \cdots, x_i \cdots, x_n) \in \mathbb{X}^n$ and $X' = (x_1, \cdots, x_i' \cdots, x_n) \in \mathbb{X}^n$, the adversary's observation (including the protocol output and any leakage), denoted by $\mathcal{A}(X)$ and $\mathcal{A}(X')$, should be close in distribution. Formally, we have the following theorem:

THEOREM 4.1 (DIFFERENTIAL PRIVACY GUARANTEE). *In the protocol given in Algorithm 1, consider neighboring datasets $X \sim_i X'$ with $i \notin C$. Define a random variable/distribution V_ϵ as:*

$$V_\epsilon^{x_i, x_i'} = \frac{\mathbb{P}[\mathcal{R}_i(x_i) = y] - e^\epsilon \cdot \mathbb{P}[\mathcal{R}_i(x_i') = y]}{\mathbb{P}[\hat{\mathcal{R}}_*(1) = y]},$$

where $y \sim \hat{\mathcal{R}}_*(1)$. Let $n^* = n - |C \cap [n]|$, then for any $\epsilon \geq 0$, we have:

$$D_\epsilon(\mathcal{A}(X) \| \mathcal{A}(X')) \leq \mathop{\mathbb{E}}_{v_1, \ldots, v_{n^* \lceil m \rceil + 1} \sim V_\epsilon^{x_i, x_i'}} \left[[\sum_{j \in [n^* \cdot \lceil m \rceil]} v_j]_+ \right].$$

The proof follows a technique similar to the privacy blanket framework [7], where dummy messages serve as "blanket" messages that help conceal the victim's message. The complete proof is provided in Appendix E.1. It proceeds in four steps: (i) simplify the privacy analysis by (safely) ignoring messages from corrupted users; (ii) further simplify the analysis by ignoring the input-dependent messages generated by other honest users (except the victim i); (iii) characterize the distribution of the shuffled input-dependent messages $\mathcal{R}_i(x_i)$ together with the dummy messages $\mathcal{R}_*(1)$ contributed by the remaining users; and (iv) express the resulting hockey-stick divergence in terms of the random variable V_ϵ.

Once the distribution $V_\epsilon^{x_i, x_i'}$ is obtained, the distribution of the summation of $n^* \cdot \lceil m \rceil$ variables drawn from $V_\epsilon^{x_i, x_i'}$ can be efficiently computed using fast Fourier transformation in $\tilde{O}(n \cdot \lceil m \rceil)$ time using $\tilde{\Omega}(n \cdot \lceil m \rceil)$ memory [45, 47].

4.4.2 Differential Privacy Guarantees for Shifted Randomizers. We note that the privacy guarantee holds for fixed x_i, x_i', while a remaining question is the (upper bound) privacy guarantee for any possible $x_i, x_i' \in \mathbb{X}$. We consider a special but common class of randomizers that have the following two properties:

- *shift property.* the nearby domain probability function $f(x, y)$ is a simple shift of template function $f(\vec{0}, y)$ and $f(x, y) = f(\vec{0}, y - x)$ holds for all possible x, y;
- *half-support property.* the support of $f(x, *)$ for any x is no more than half of the domain $\mathbb{Y}$. That is $\int_y [\![f(x, y)]\!] \mathrm{d}y \leq V(\mathbb{Y})/2$ holds for all possible x.

In our protocols for LBS, such two properties often hold: (i) the distribution over the obfuscating area is often uniform-random or follow a specific distribution; (ii) the obfuscating area is relatively small to ensure no failure probability. Under these assumptions, we can show that the divergence of any possible x_i, x_i' is upper bounded by the case where the supports of $\mathcal{R}_i(x_i)$ and $\mathcal{R}_i(x_i')$ do not overlap with each other, and the privacy upper bound can be simplified to Theorem 4.2 (see proof in Appendix E.2).

Theorem 4.2 (Differential Privacy Upper Bound for Shifted Randomizers). *For a local randomizer satisfies the* shift property *and* half-support property*, define a random distribution V_ϵ' as:*

$$V_\epsilon' = \begin{cases} V(\mathbb{Y}_i)\cdot f(\vec{0},y), & w.p.\ \frac{m}{\lceil m\rceil V(\mathbb{Y}_i)}\ \text{for } y\in S_{\vec{0}};\\ -e^\epsilon\cdot V(\mathbb{Y}_i)\cdot f(\vec{0},y), & w.p.\ \frac{m}{\lceil m\rceil V(\mathbb{Y}_i)}\ \text{for } y\in V_{\vec{0}};\\ 0, & w.p.\ 1-2\cdot\frac{V(S_{\vec{0}})}{V(\mathbb{Y}_i)}\cdot\frac{m}{\lceil m\rceil}, \end{cases} \tag{5}$$

where $S_{\vec{0}} = V(\{y \mid f(\vec{0},y) > 0\})$ is the support volume of $f(\vec{0},*)$. Then for any neighboring datasets $X \sim_i X'$ that $i \notin C$, the protocol satisfies (ϵ,δ)-DP for any $\epsilon \geq 0$ with

$$\delta = \mathop{\mathbb{E}}_{v_1,\ldots,v_{n^*\lceil m\rceil}\sim V_\epsilon'}\left[\left[\sum_{j\in[n^*\cdot\lceil m\rceil]} v_j\right]_+\right].$$

4.4.3 Differential Privacy Guarantees for Uniform Randomizers. Another special randomizer maps x to an output sampled uniformly at random from $\mathbb{Y}_i(x)$, i.e., $f(x,z) = f(x,z')$ for all $z,z' \in \mathbb{Y}_i(x)$. If, in addition, the randomizer satisfies the *half-support* property and the support volume is identical for all possible inputs, then the hockey-stick divergence admits an explicit numerical characterization. Specifically, Theorem 4.3 (proved in Appendix E.3) provides a computable expression that can be evaluated in $\tilde{O}(n\lceil m\rceil)$ time and $O(1)$ memory.

Theorem 4.3 (Differential Privacy in Numerical Form for Uniform Randomizers). *For any uniform randomizer $\mathcal{R}_i$ has input-independent support volume for all possible input (i.e., $V(\mathbb{Y}_i(x)) \equiv V(\mathbb{Y}_i)$), for neighboring datasets $X \sim_i X'$ that $i \notin C$, the protocol satisfies (ϵ,δ)-DP for any $\epsilon \geq 0$ and*

$$\delta = \mathop{\mathbb{E}}_{s\sim Binom(n^*\lceil m\rceil,2t)}\left[\underset{s,1/2}{\mathsf{CDF}}[low_{s+1}-1,s] - e^\epsilon \underset{s,1/2}{\mathsf{CDF}}[low_{s+1},s]\right],$$

where $t = \frac{m}{\lceil m\rceil}\cdot\frac{V(\mathbb{Y}_i)}{V(\mathbb{Y})}$, $low_s = \frac{e^\epsilon(s+1)}{e^\epsilon+1}$, and $\mathsf{CDF}_{s,1/2}[a,b]$ denotes the cumulative probability of binomial distribution Binomial$(s,1/2)$ over the range $[a,b]$.

Theorem 4.3 shows that, given ϵ, the corresponding δ can be written as an expectation of binomial cumulative probabilities. Hence, δ can be evaluated in $\tilde{O}(n^*\lceil m\rceil)$ time. As detailed in Appendix E, the proof proceeds in four main steps: (i) simplify the analysis by (safely) ignoring messages from corrupted users, and also ignoring input-dependent messages from other honest users; (ii) upper bound the divergence by considering only the non-overlapping support case (similar to Theorem 4.2); (iii) motivated by the clone reduction technique [21, 23], reduce the divergence to a comparison between dominating binomial counts in non-overlapping regions; (iv) inspired by the numerical method of [55], exploit the monotonicity of the Hockey-stick divergence with respect to binomial counts to derive the final expectation form.

Compared with existing works [22, 23, 45, 47, 55], which analyze identically randomized reports with a fixed number of messages, our analysis deals with non-LDP randomizers and a random number of dummy messages per user (since the dummy count m' is randomized whenever $m \neq \lceil m\rceil$).

Asymptotically, when n^* is sufficiently large, the privacy loss ϵ can be approximated by $O(\sqrt{\log(1/\delta)V(\mathbb{Y})/(n^*mV(\mathbb{Y}_i))})$ (see Proposition 4.4, proved in Appendix E.4). This implies that setting the number of dummy messages per user as

$$m = O(\log(1/\delta)V(\mathbb{Y})/(n^*\epsilon^2V(\mathbb{Y}_i))) \tag{6}$$

is enough to achieve (ϵ,δ)-DP in the protocol.

Proposition 4.4 (DP in Closed Form). *Under the same assumption as Theorem 4.3, for neighboring datasets $X \sim_i X'$, when $n^*mt' \geq 16\log(4/\delta)$, the protocol satisfies (ϵ,δ)-DP with:*

$$\epsilon = \log\left(1 + 16\sqrt{\frac{\log(4/\delta)}{3n^*mt'}} + \frac{8}{3n^*mt'}\right), \tag{7}$$

where $t' = V(\mathbb{Y}_i)/V(\mathbb{Y})$.

For small n^* (e.g., $n = 500$), we determine m numerically using Theorem 4.3 rather than the closed-form Eq. (6).

Privacy lower bounds. The numerical privacy bound in Theorem 4.3 is tight under mild assumption on the uniform randomizers (i.e., half supports and identical supporting volumes). In particular, there exist two neighboring datasets such that the protocol's Hockey-stick divergence δ equals to the one in Theorem 4.3 (see Proposition 4.5 for a formal statement, and Appendix E.5 for proof).

Proposition 4.5 (Privacy Lower Bound of Uniform Randomizers). *Under the same assumption as Theorem 4.3, let $t = \frac{m}{\lceil m\rceil}\cdot\frac{V(\mathbb{Y}_i)}{V(\mathbb{Y})}$, $low_s = \frac{e^\epsilon(s+1)}{e^\epsilon+1}$. There exist two neighboring datasets $X \sim_i X'$ and an algorithm $\mathcal{A}$, such that:*

$$\max\{D_\epsilon(\mathcal{A}(X)\|\mathcal{A}(X')), D_{\epsilon_c}(\mathcal{A}(X')\|\mathcal{A}(X))\}$$
$$\geq \mathop{\mathbb{E}}_{s\sim Binom(n^*\cdot\lceil m\rceil,2t)}\left[\underset{s,1/2}{\mathsf{CDF}}[low_{s+1}-1,s] - e^\epsilon \underset{s,1/2}{\mathsf{CDF}}[low_{s+1},s]\right].$$

4.5 Utility Analyses

Service utility guarantees. In many LBS tasks (e.g., navigation or location-based information retrieval), service quality largely depends on the utility of the input-dependent report. For honest users, the utility guarantee of our design is straightforward to see. Specifically, recall that r_* denotes the universal service failure radii, if $r_i \leq r_*$, then the support $\mathbb{Y}_i(x)$ of the input-dependent message $\mathcal{R}_i(x)$ is contained in the success range $\mathbb{B}_{r_*}(x)$. As a result, the service failure probability is zero (see Theorem 4.6).

Theorem 4.6 (Zero Service Failure Guarantee). *In the Ploda protocol, if $\mathbb{Y}_i(x) \subseteq \mathbb{B}_{r_*}(x)$, then for any $x \in [-1,1]^2$, we have $\mathbb{P}[\|\mathcal{R}_i(x) - x\|_2 \geq r] = 0$.*

Proof. Recall that $\mathbb{B}_r(x) = \{y \mid y \in \mathbb{R}^2 \text{ and } \|y-x\|_2 < r\}$, if $\mathbb{Y}_i(x) \subseteq \mathbb{B}_{r_*}(x)$, then we have $\mathbb{P}[\mathcal{R}_i(x) \notin \mathbb{B}_r(x)] = 0$, which implies that $\mathbb{P}[\|\mathcal{R}_i(x) - x\|_2 \geq r] = 0$. □

Aggregation utility guarantees without corrupted users. An aggregation task aims to estimate statistics over a region $a \subseteq$

$[-1,1]^2$, such as the number of users within a service block. A natural estimator counts the number of observed messages falling in a, and then subtracts the expected contribution of dummy messages. Specifically, in a benevolent environment where $C \cap [n] = \emptyset$, suppose the protocol produces in total M dummy messages (i.e., the number of observed messages minus the number of users). Since each dummy is sampled uniformly from $\mathbb{Y}$, the expected number of dummy messages that fall inside a equals $M \cdot \frac{\mathrm{V}(a)}{\mathrm{V}(\mathbb{Y})}$. Therefore, given the shuffled multiset of received messages L, the server estimates the number of users in a by

$$\hat{N}_a = \#\{z \mid z \in L, z \in a\} - (|L| - n)\frac{\mathrm{V}(a)}{\mathrm{V}(\mathbb{Y})}. \tag{8}$$

The variance induced by the dummy messages is

$$M \cdot \frac{\mathrm{V}(a)}{\mathrm{V}(\mathbb{Y})} \cdot \left(1 - \frac{\mathrm{V}(a)}{\mathrm{V}(\mathbb{Y})}\right) = O\left(\frac{\log(1/\delta)}{\epsilon^2} \cdot \frac{\mathrm{V}(a)}{\mathrm{V}(\mathbb{Y}_i)}\right),$$

which matches the same scale as in the curator DP model using Gaussian noise.

It should be noted that there can be estimation errors caused by *bounce-out* users (whose true location x lies in a but whose obfuscating domain $\mathbb{Y}_i(x)$ extends outside a) and *bounce-in* users (whose true location x lies outside a but whose obfuscating domain $\mathbb{Y}_i(x)$ overlaps with a). Fortunately, this is not an issue when points of interest are well separated and the obfuscating domain $\mathbb{Y}_i(x)$ is smaller than half of the location separation. In this case, the server can map each reported location to the nearest point of interest, which eliminates bounce-in and bounce-out reports.

Aggregation utility guarantees with corrupted users. When there are $|C \cap [n]|$ corrupted users, they not only affect the protocol parameter m required to ensure (ϵ, δ)-DP for honest users, but may also introduce bias into the estimated statistics. In particular, the expected number of dummy messages per user is $O\left(\frac{\log(1/\delta)\,\mathrm{V}(\mathbb{Y})}{(n-|C\cap[n]|)\epsilon^2\,\mathrm{V}(\mathbb{Y}_i)}\right)$. Assume the well-separated setting where the obfuscating radius is at most half of the gap between discrete locations, so that bounce-in and bounce-out errors do not occur. Then the estimation error is dominated by two sources: (i) the randomness of uniform dummy messages, and (ii) the arbitrary messages contributed by users in $C \cap [n]$. For sufficiently large n, we have:

$$\begin{aligned}
&\mathbb{E}[|\hat{N}_a - N_a|_1] \\
\le& O\left(\sqrt{\frac{\log(1/\delta)V(\mathbb{Y})}{\epsilon^2 V(\mathbb{Y}_i)}\frac{V(a)}{V(\mathbb{Y})}}\right) + (n - n^*) \cdot O\left(\lceil \frac{\log(1/\delta)V(\mathbb{Y})}{n^*\epsilon^2 V(\mathbb{Y}_i)} \rceil\right) \\
\le& O\left(\sqrt{\frac{\log(1/\delta)}{\epsilon^2} \cdot \frac{\mathrm{V}(a)}{\mathrm{V}(\mathbb{Y}_i)}} + |C| \cdot \lceil \frac{\log(1/\delta)V(\mathbb{Y})}{(n - |C|)\epsilon^2 V(\mathbb{Y}_i)} \rceil\right).
\end{aligned}$$

Specifically, under per-user rate limitation of $\lceil m \rceil$, the maximum absolute error/bias contributed by corrupted messages to location counts is $|C| \cdot (1 + \lceil m \rceil)$. This is only a factor of $(1 + \lceil m \rceil)$ larger than the corresponding bias in the non-private setting. Since $m = \tilde{O}(\log(1/\delta)/(n^*\epsilon^2) \cdot V(Y)/V(Y_i))$ becomes small when n^* is relatively large, we typically have $m \le 1$ in such regimes and the maximum bias is at most twice that of non-private setting.

Remark 4.3 (Ablation of protocol components). Ploda combines three components, each addressing a distinct requirement: bounded obfuscation, dummy messages, and shuffling. From a privacy perspective, no single component, nor any pair of them, provides meaningful DP guarantee; DP emerges only from their joint use. From a utility perspective, the bounded obfuscation ensures zero service failure; dummy messages are the primary source of randomness in the privacy-amplification analysis, introducing the variance required for DP; and shuffling breaks the user-message linkage, which both enables dummies to provide privacy amplification and allows dummy messages to be shared across users, explaining why Ploda achieves low message complexity per user and correspondingly low dummy-induced variance (see Section 4.5).

Remark 4.4 (Model positioning between local and central DP). Local DP assumes no trusted party but incurs the service-failure lower bounds of Section 3.3. Central DP assumes a fully trusted curator that observes all raw locations. Ploda's shuffler model lies strictly between these two: under a split-trust deployment, no single non-colluding party observes both sender identity and plaintext location. Concretely, user messages can be encrypted under the server's public key, and the server's per-message responses can be encrypted under one-time keys attached to each message, following the PIC shuffle model [51]; the shuffler then sees only ciphertexts, while the server sees only the shuffled multiset. The shuffler/server or a third party might also sees additional ciphertexts used for anonymous rate limitation. If all these roles are collapsed into a single entity, or if the shuffler and server collude, the model can degenerate to central DP.

5 Personalized Privacy-Utility Tradeoffs

In many LBS services, users might have diverse service failure radius and/or privacy requirements. In practice, *utility-focused users* (e.g., Uber drivers, couriers, ambulances) may adopt a small nearby radii $r_i \in (0, \hat{r}]$, while *utility-insensitive users* (e.g., social app users, fitness trackers) may prefer a larger nearby radii $r_i \in (0, \hat{r}]$. In our protocol, each user's nearby radii r_i determines the input-dependent obfuscating domain $\mathbb{Y}_i(x_i) = \{y \mid \|y - x_i\|_2 < r_i\}$ and the privacy guarantees (see Section 4.4), a smaller r_i implies more privacy consumption. This creates a flexible *privacy-utility tradeoff* for each individual user, where each user can choose their own radii r_i.

To formalize heterogeneous privacy guarantees, we adopt personalized DP [36], where each user $i \in [n]$ has their own privacy parameter $\epsilon_i \ge 0$. For neighboring datasets $X, X' \in \mathbb{X}^n$ differing only at the i-th entry (denoted $X \sim_i X'$), the privacy guarantee is defined through output indistinguishability (we assume the δ parameter is the same for all users for simplicity):

Definition 5.1 (Personalized Differential Privacy). *Let $\epsilon_1, \ldots, \epsilon_n \ge 0$ be user-specified privacy parameters and $\delta \in [0,1]$. A mechanism $\mathcal{M} : \mathbb{X}^n \to \mathbb{Z}$ satisfies $(\{\epsilon_i\}_{i=1}^n, \delta)$-personalized differential privacy if for every user $i \in [n]$ and all neighboring datasets $X \sim_i X'$, the output distributions satisfy that $\mathcal{M}(X)$ and $\mathcal{M}(X')$ are (ϵ_i, δ)-indistinguishable.*

Our protocol naturally supports personalized obfuscating radii r_i and personalized privacy guarantee ϵ_i. The key observation is that in our protocol, the privacy guarantee for each user is provided by the dummy messages, which are input-independent and identically distributed across all users. Consequently, in Section 4.4, the privacy analysis for user i depends only on their own input-dependent

randomizer $\mathcal{R}_i$ and the collective dummy messages, but *not* on the input-dependent randomizers of other users. This means our privacy analyses in Theorems 4.1, 4.2, 4.3 and Proposition 4.4 still hold when each user chooses a different nearby radius r_i and a different $\mathcal{R}_i$. In particular, variations in other users' obfuscating domains and input-dependent randomizer have no *negative* effect on the user i's privacy. Moreover, under a mild non-overlapping assumption on the supports of input-dependent randomizers, they provide no *positive* effect either, meaning that our personalized privacy analyses can be tight. This is formalized in Proposition 5.2.

Proposition 5.2. *In the protocol of Algorithm 1, for neighboring dataset relation $X \sim_i X'$ that $i \notin C$, we have:*

$$D_\epsilon(\mathcal{S}(Z_1 \cup \ldots \cup Z_i \cup \ldots \cup Z_n \| Z_1 \cup \ldots \cup Z_i' \cup \ldots \cup Z_n))$$
$$\leq D_\epsilon\Big(\mathcal{S}(\mathcal{R}_i(x_i) \cup (\bigcup_{j\in[n]\setminus C} \mathcal{R}_*(m))) \| \mathcal{S}(\mathcal{R}_i(x_i') \cup (\bigcup_{j\in[n]\setminus C} \mathcal{R}_*(m)))\Big).$$

When there are inputs $\{x_j\}_{j\in[n]\setminus(C\cup\{i\})}$ such that both $\mathbb{Y}_i(x_i) \cap (\cup_{j\in[n]\setminus(C\cup\{i\})} \mathbb{Y}_j(x_j)) = \Phi$ and $\mathbb{Y}_i(x_i') \cap (\cup_{j\in[n]\setminus(C\cup\{i\})} \mathbb{Y}_j(x_j)) = \Phi$ hold, then there exists two neighboring datasets $X \sim_i X'$ such that:

$$D_\epsilon(\mathcal{S}(Z_1 \cup \ldots \cup Z_i \cup \ldots \cup Z_n) \| \mathcal{S}(Z_1 \cup \ldots \cup Z_i' \cup \ldots \cup Z_n))$$
$$\geq D_\epsilon\Big(\mathcal{S}(\mathcal{R}_i(x_i) \cup (\bigcup_{j\in[n]\setminus C} \mathcal{R}_*(m))) \| \mathcal{S}(\mathcal{R}_i(x_i') \cup (\bigcup_{j\in[n]\setminus C} \mathcal{R}_*(m)))\Big).$$

6 Numerical Experiments

In this part, we evaluate the performances of our proposal, with comparison to classical location privacy preserving paradigms.

6.1 Settings

Datasets. We use four real-world datasets: Gowalla [12], Geolife [62], Portocabs[1], and Foursquare [58]. The location data in these datasets are normalized to $[-1,1]\times[-1,1]$. The service radius r_* (in ℓ_2 distance) varies from 0.05 to 0.2, to simulate extensive scenarios. We describe them in what follows:

- **Gowalla** is a location-based social network dataset with user check-ins collected between February 2009 and October 2010, covering over 100,000 users and 6.4 million check-ins worldwide. Each record includes a user ID, timestamp, and latitude–longitude coordinates.
- **Geolife** is a GPS trajectory dataset collected by Microsoft Research Asia over five years (2007–2012), comprising more than 17,000 trajectories, 1.2 million kilometers of travel, and 25 million location points.
- **Portocabs** is a taxi trajectory dataset collected in Porto, Portugal over one year, containing approximately 1.7 million trajectories and over 150 million timestamped GPS points.
- **Foursquare** contains location-based social networking check-ins from New York (Foursquare NYC) and Tokyo (Foursquare TKY), with millions of records including venue categories, timestamps, and precise geospatial coordinates, enabling studies in dense urban environments.

Methods. We compare our Ploda protocol (uses uniform obfuscating over $\mathbb{B}_{r_*}(x_i)$) with the following:

[1]https://figshare.com/articles/dataset/Porto_taxi_trajectories/12302165/1

- **Laplace mechanism in LDP.** Each user adds two-dimensional Laplace noises to the true location with sensitivity 4.
- **PlanarLaplace mechanism in GI.** Each user adds PlanarLaplace noises to the true location. For fair comparison with other DP approaches, we restrict the GI's maximum distinguishability level over domain $[-1,1]^2$ at ϵ (i.e., ϵ-LDP).
- **Minkowski response in LDP.** Minkowski response mechanism [51] with cap area fits the ℓ_2 service radius r_*.
- **Laplace mechanism in PIC shuffle model.** Add Laplace noises to the true location with sensitivity 4, and with shuffle-amplified local privacy budget [21].
- **Minkowski response in PIC shuffle model.** Minkowski response mechanism with cap area fits the ℓ_2 service radius r_*, and with shuffle-amplified local privacy budget [21].

Evaluation metrics. We compare the average service failure ratios, location distribution/count estimation errors, and message complexities of these approaches. We present the averaged measures of 200 repeated experimental simulations. Specifically, the metrics are defined as follows:

- **Service Failure Ratio** is the expected ratio that location reports exceed a predefined spatial radius from the true locations: $P_{fail} = (\sum_{i=1}^n \mathbb{I}(\|x_i - y_i\|_2 \geq r_*))/n$, where $\|x_i - y_i\|_2$ is the ℓ_2 distance between the true location and the input-dependent location report, $\mathbb{I}(\cdot)$ is the 0/1 indicator function.
- **Average distance** is the average ℓ_2 distance $\|y_i - x_i\|_2$ between the true location and the input-dependent report.
- **Location Distribution Error** is the error between the true and estimated location distributions or range counts. These distributions are normalized by the total number of users n.
- **Message Complexity** is the expected number of messages transmitted between a user and service provider. For all approaches except Ploda, the message complexity is 1.

6.2 Location-based Retrieval Services

6.2.1 Service failures. In location information retrieval, we compare the average service failures of existing approaches. We note that Ploda protocol will always incur no service failure. We also note that the average failure ratio deviates from the minimax failure probabilities derived in Section 3.2, as the results now depend on the concrete true locations (instead of worst-case inputs) and the concrete LDP/GI mechanisms (instead of best mechanisms).

Figure 2 presents the service failure probabilities across different privacy budgets ϵ. As expected, Ploda achieves zero failure probability regardless of ϵ, while all other approaches exhibit non-trivial failure rates. For example, in the typical setting of $\epsilon = 1$, the Laplace mechanism in LDP incurs over 99.8% average failure rate, while the Minkowski mechanism in LDP at 94%. The shuffle-amplified approaches (Laplace and Minkowski in PIC shuffle model with $n = 5,000$) reduce failure rates to 95% and ~26% respectively, but still cannot match Ploda's perfect service reliability. Increasing ϵ reduces failure rates for all mechanisms, but even at $\epsilon = 4$, the best competitor (Minkowski in PIC) still has >10% failure rate.

6.2.2 Average distance. Figure 3 shows the average distance between true locations and reports. Due to the design of its input-dependent message, the average distance of Ploda will be fixed at $2r_*/3$, regardless of the privacy level. Under $\epsilon = 1$ and $r = 0.1$, the

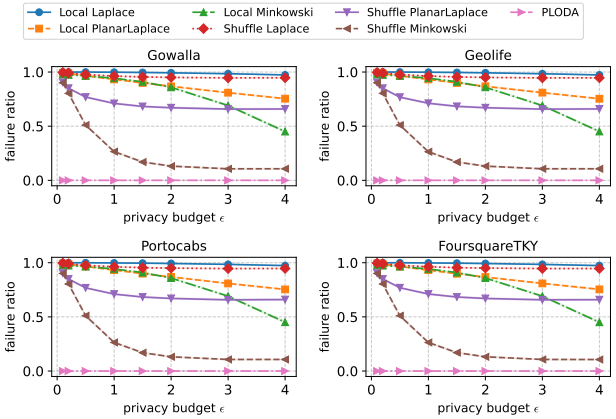

Figure 2: Service failure probabilities vs. privacy budget ϵ for $r_* = 0.2$ and $n = 5,000$.

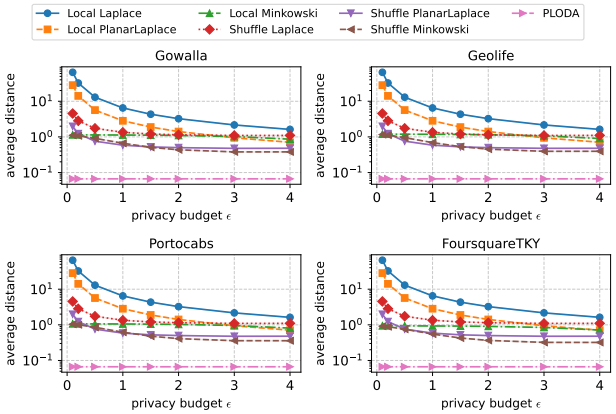

Figure 3: Average distance vs. privacy budget for $r_* = 0.1$ and $n = 5,000$.

Ploda has average distance of 0.067, while the Laplace mechanism in LDP has 6.4, the Minkowski mechanism in LDP has 0.92. Even with privacy amplification in PIC, the Laplace mechanism in LDP has 1.3, the Minkowski mechanism in LDP has 0.54, demonstrating Ploda's superior accuracy for serviceable reports.

6.2.3 Impact of user counts. Figures 4 and 5 present service failure and average report distance results when $n = 1,000$. Compared to Figures 2 and 3, the results in LDP and Ploda are the same, while these in the PIC shuffle model performs worse, as the privacy amplification effects become weaker.

6.2.4 Message complexities. Our Ploda protocol trades message complexities for service failures. Table 3 presents the required message complexity per user for different numbers of users n and privacy budgets ϵ. For practical settings ($n = 1,000$, $\epsilon = 1$, $r_* = 0.2$), Ploda requires only 1.625 dummy messages per user, demonstrating its efficiency. As n increases, message complexity decreases dramatically, becoming negligible for large user bases (≈ 0.2 messages per user for $n = 10^4$).

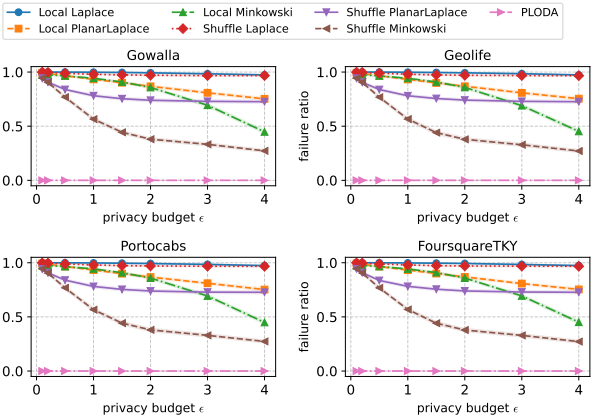

Figure 4: Service failure probabilities vs. privacy budget ϵ for $r_* = 0.2$ and $n = 1,000$.

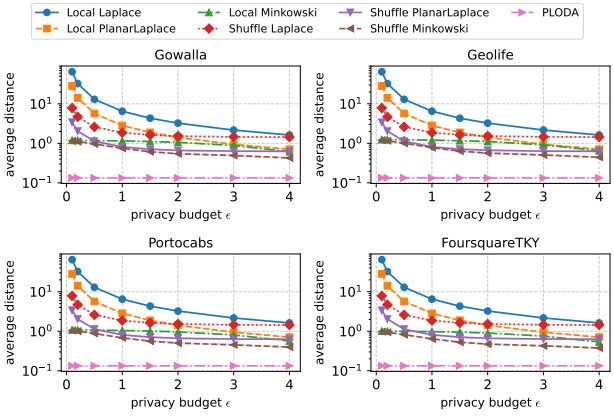

Figure 5: Average distance for $r_* = 0.2$ and $n = 1,000$.

Table 3: Message complexity per user in typical settings with $\delta = 0.01/n$.

n	ϵ	dummies ($r_* = 0.1$)	dummies($r_* = 0.2$)
500	0.5	34.0	10.0
500	1.0	11.0	3.25
1,000	0.5	17.0	5.0
1,000	1.0	5.5	1.625
5,000	0.5	4.0	1.188
5,000	1.0	1.25	0.375
10,000	0.5	2.125	0.625
10,000	1.0	0.6875	0.2032

6.3 Location Distribution Estimation

We consider the case where service providers analyze service usages (e.g., user location distribution) in LBS. We assume gridded rectangular location areas of interest and aim to estimate user counts in each area. For Laplace/PlanarLaplace mechanisms, we estimate distributions by counting reports in each area. For Minkowski response and Ploda, we use debiased count estimators (as in [10, 50]). In such cases, the Minkowski response mechanism is equivalent to

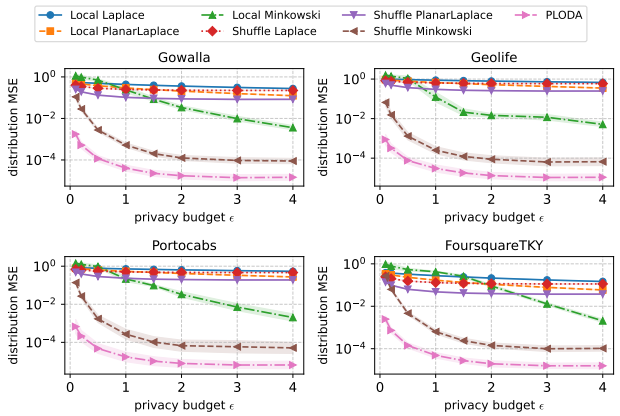

Figure 6: Location distribution estimation MSE vs. privacy level for 5×5 grid areas (i.e. $r_* = 0.2$) and $n = 5000$.

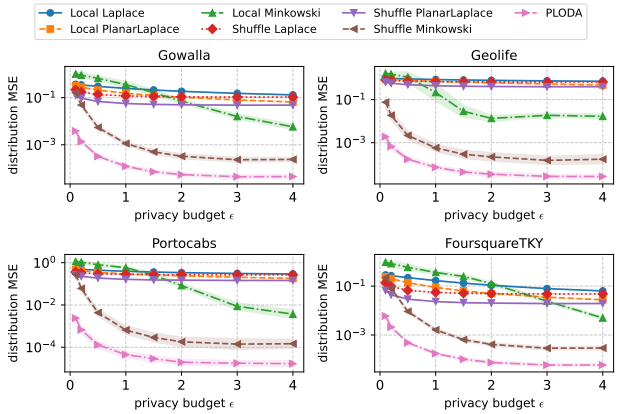

Figure 7: Location distribution estimation MSE vs. ϵ for 10×10 grid areas ($r_* = 0.1$) and $n = 5000$.

the generalized randomized response mechanism [56], an optimal mechanism in the single-message shuffle model [21].

6.3.1 Estimation errors. Figure 6 shows estimation MSE for 5×5 grid areas. PLODA achieves significantly lower MSE than all competitors, demonstrating its superior utility-preserving properties. For example, at $\epsilon = 1$, PLODA's MSE is 0.000048, compared to 0.00063 for the most competitive Minkowski in PIC shuffle model, reducing over 90% error. Figures 7 and 10 show similar results for 10×10 and 20×20 grids respectively, with PLODA maintaining its advantage across all privacy budgets.

6.3.2 Varying user counts. Figures 8 and 9 demonstrate the estimation error with $n = 1,000$ and $n = 10,000$, respectively. PLODA's MSE decreases rapidly with n, achieving large performance advantages with $n = 10,000$ over other approaches (compared to the cases of $n = 1,000$ or $n = 5,000$). This phenomenon validates our theoretical guarantees in Theorem 4.5, which shows PLODA achieves the same error rate on n as the curator DP mechanisms, while local DP and PIC shuffle model are unable to.

6.3.3 Range queries. We evaluate range query accuracy based on the location distribution, over a 10×10 grid with $r^* = 0.1$ and $n = 5{,}000$ (Figure 11). A range query requests the total user count

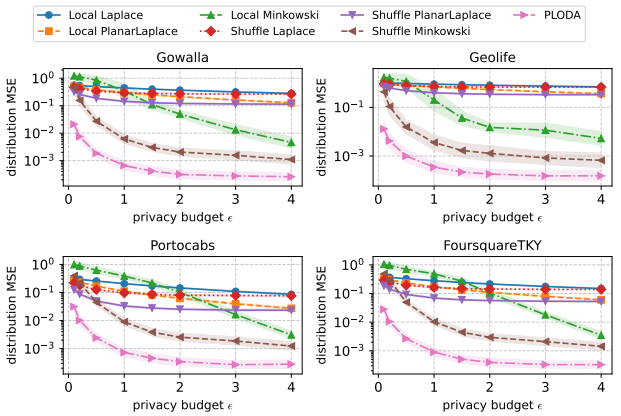

Figure 8: Location distribution estimation MSE vs. privacy level for 5×5 grid, $r^* = 0.2$, and $n = 1,000$.

within a rectangular sub-grid $[i_1, i_2] \times [j_1, j_2]$. To eliminate query-selection variance, we pre-generate 200 random rectangular queries per privacy budget and repeat, and evaluate all mechanisms on the same query set. We report the mean absolute error (MAE) between the true and estimated normalized counts. PLODA achieves lower MAE than all LDP and PIC-shuffle baselines across privacy budgets, consistent with its lower distribution-estimation error.

6.3.4 Hotspot identification. We evaluate how well each mechanism preserves spatial hotspots on 10×10 ($r^* = 0.1$) and 20×20 ($r^* = 0.05$) grids with $n = 5{,}000$ (Figures 12 and 13). A cell is flagged as a hotspot when its user count exceeds 5% of n^*. We compute the F1 score by comparing the predicted hotspot set against the true hotspot set derived from the original distribution; since the two sets may differ in cardinality, precision and recall are reported separately and combined into F1. PLODA achieves higher F1 (i.e., near 100% score even under low budget) than the LDP and PIC-shuffle baselines, indicating that its lower estimation error better preserves the hotspot structure.

6.3.5 Poisoning attacks. We evaluate robustness against data poisoning, where corrupted users inject adversarial messages to bias aggregate statistics (Figures 14 and 15). We vary the corrupted-user count $|C|$ from 0 to 10% of n^* on a 10×10 grid ($r^* = 0.1$, $n = 5{,}000$), where each corrupted user injects up to $\lceil m \rceil + 1$ messages targeting a single cell, under $\epsilon \in \{0.5, 1.0\}$. We report the range-query MAE as $|C|$ grows. Consistent with our analysis in Section 4.5, the aggregate utility degrades gracefully with $|C|$; the per-user rate limit bounds the resulting bias.

7 Related Work

Preserving location privacy has been extensively studied due to the pervasive usage of location data and its sensitive nature. We survey existing approaches through two paradigms: classical methods and modern differential privacy techniques.

7.1 Classical Location Privacy Preservation

Classical techniques employ intuitive perturbation strategies. A straightforward approach is replacing a fine-grained location with a coarse-grained one, known as generalization [26, 46]; another

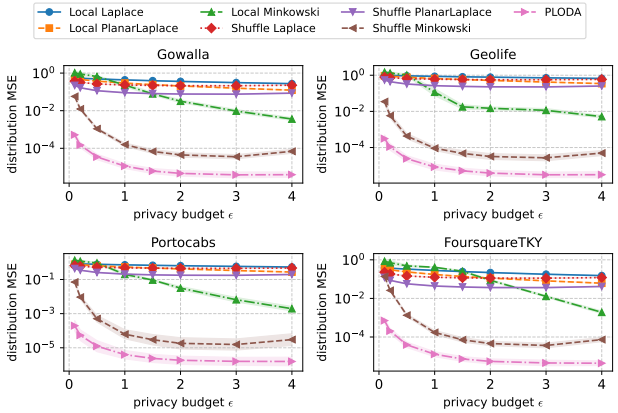

Figure 9: Location distribution estimation MSE vs. privacy level for 5×5 grid and $n = 10,000$.

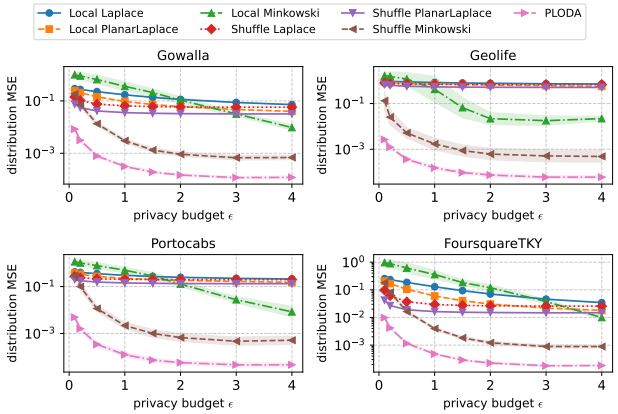

Figure 10: Location distribution estimation MSE vs. privacy level for 20×20 grid, $r^* = 0.05$, and $n = 5,000$.

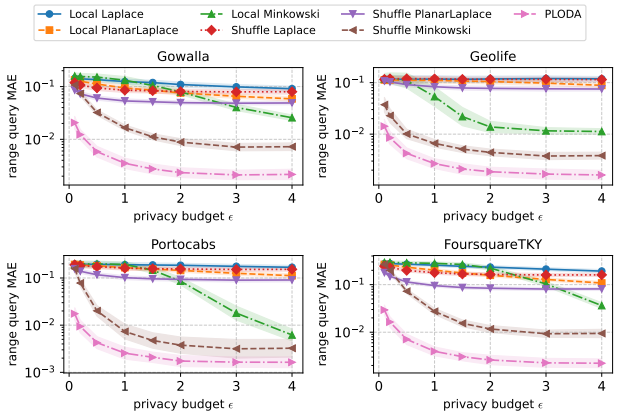

Figure 11: Location range count MAE vs. privacy level for 10×10 grid, $r^* = 0.1$, and $n = 5,000$.

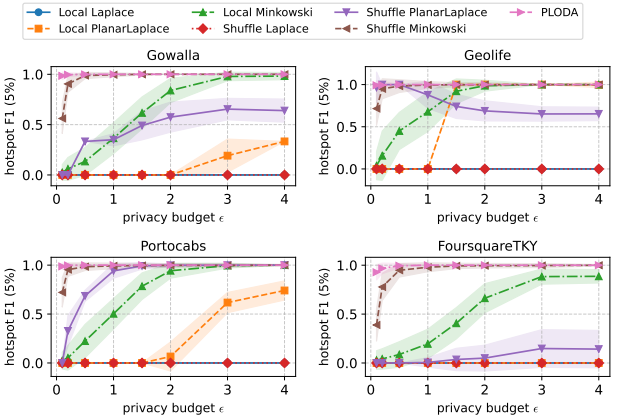

Figure 12: Location hotspots (over $5\% \cdot n^*$ count) identification F1 score vs. privacy level for 10×10 grid, $r^* = 0.1$, and $n = 5,000$.

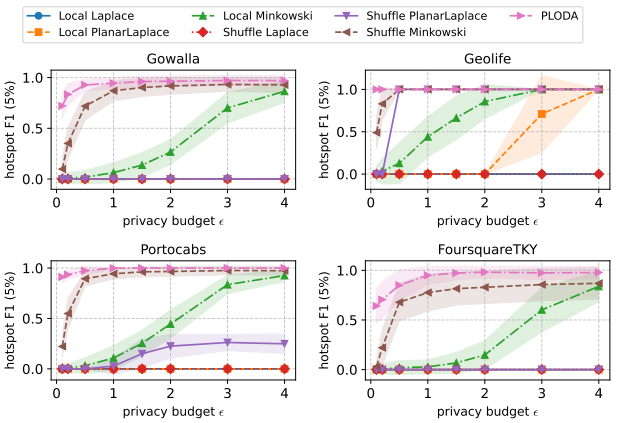

Figure 13: Location hotspots (over $5\% \cdot n^*$ count) identification F1 score vs. privacy level for 20×20 grid, $r^* = 0.05$, and $n = 5,000$.

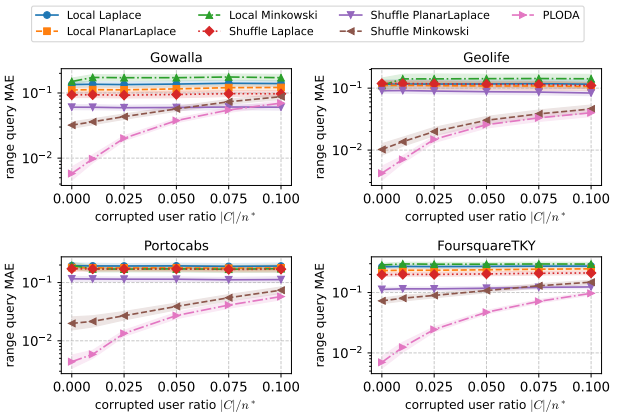

Figure 14: Location range query MAE vs. number of corrupted users, for 10×10 grid, $r^* = 0.1$, $\epsilon = 0.5$, and $n^* = 5,000$.

simple approach perturbs exact locations to nearby random points, known as location obfuscating [4, 5]. Anonymization methods dissociate user identities from locations [26, 29]. Researchers have also proposed supplementing true locations with dummy reports [40, 57] to obscure genuine positions. Despite offering implementation simplicity and being widely adopted in mobile location services, these approaches prove vulnerable to inference attacks, leading to significant individual location exposure rates [61].

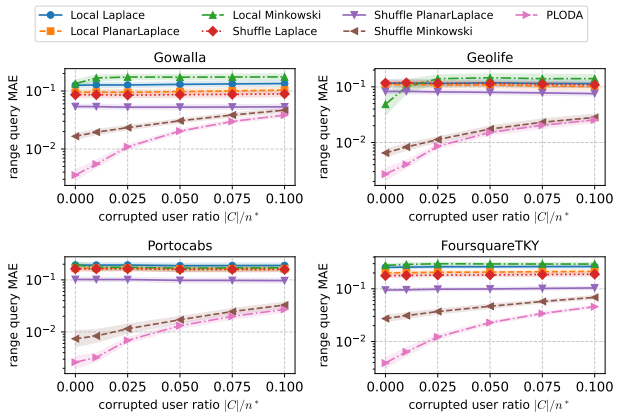

Figure 15: Location range query MAE vs. number of corrupted users, for 10×10 grid, $r^* = 0.1$, $\epsilon = 1.0$, and $n^* = 5,000$.

7.2 Differential Privacy for Locations

Differential privacy [19] provides formal guarantees against strong adversaries. For location data, a series of works preserve location privacy in the curator model of DP, such as for distribution estimation [14] and location trace synthesis [30]. In the local DP model that naturally aligns with decentralized LBS architectures, [41, 50] employ randomized response mechanisms for point-of-interest reporting, [10] permits personalized LDP level for location privacy. The geo-indistinguishability variant [2] enhances utility by calibrating noise to geographical distance, typically implemented through planar Laplace mechanisms [2] or designated transition probabilities [52, 53]. However, these approaches incur an inherent service failure tradeoff: satisfying DP constraints requires non-zero probability across the entire domain, causing significant deviations that disrupt services. Our work addresses this tension between formal privacy and service success.

8 Discussions

8.1 Sequential Composition

It is common for users to utilize location-based services *periodically*, such as for repeated location updates in navigation. Each user may participate in multiple sequential protocol runs. In this case, one can use advanced sequential composition of DP [19, 38] to derive a closed-form bound of the overall privacy loss. Specifically, [38] shows that, for k independent executions of (ϵ, δ)-DP mechanisms (e.g., fresh bounded obfuscation, fresh dummy messages, and fresh shuffling in multiple PLODA runs), the overall privacy loss is $(\epsilon', 1-(1-\delta)^k(1-\delta'))$ for any $\delta' > 0$, where $\epsilon' = \epsilon\sqrt{2k\ln(1/\delta')} + k\epsilon(e^\epsilon - 1)/(e^\epsilon + 1)$. When k is relatively large and the one-round privacy loss ϵ is relatively small (e.g., $\epsilon = \Theta(k\delta)$), the leading term is $\epsilon\sqrt{2k\ln(1/\delta')}$, and the total privacy loss asymptotically scales as $\epsilon' = \tilde{O}(\epsilon\sqrt{k})$ [38]. Alternatively, a tighter overall privacy loss can be computed using fast Fourier accounting [42] based on the (ϵ, δ)-DP curve (i.e., the divergence δ curve with varying ϵ in Theorem 4.3) of each protocol run. Moreover, for location traces that exhibit slow variation across rounds, the inherent sparsity in their temporal changes can be leveraged to further reduce the $\sqrt{k}$ factor, an approach analogous to that in [54]. We leave a detailed exploration of this extension to future work.

8.2 Relation to Differentially Private PIR

Private Information Retrieval (PIR [13]) aims to retrieve a data item from a server-side public dataset without revealing the specific item index. Besides the major line of PIR research based on information-theoretic security [34, 35] or computational security[25], there is also a line of work [1, 6, 49] that relaxes the disclosure requirement in the differential privacy sense. However, these approaches operate on a discrete index space and often require multiple servers, with significant cryptographic operations on user/server sides. Additionally, their protocols prevent the server from learning the query distribution, which can be important for improving the quality of LBS. In contrast, our PLODA protocol operates over a continuous location domain, requires minimal adaptation on both the user and server sides (compared to non-private settings), and can simultaneously support high-utility location distribution estimation.

9 Conclusion

This work studies location privacy protection from a novel service failure perspective, emphasizing the risk of location service failure due to privacy sanitation. We analyzed the intrinsic limitations of existing formal protection methods (e.g., local DP, geo-indistinguishability, and PIC shuffle model), and show that they necessarily incur significant failure probabilities. Consequently, we propose a new framework PLODA that incurs zero failure probability while preserving differential privacy. It employs classical techniques of obfuscation, dummy, and anonymization, with shuffle DP analyses. Experimental results demonstrate the supreme effectiveness and efficiency of the PLODA for location privacy protection.

Acknowledges

Shaowei Wang is supported by National Natural Science Foundation of China (No.62372120, No.62102108), Guangzhou Basic and Applied Basic Research Foundation (No.2025A03J3182), and Guangdong Provincial Association for Science and Technology (No.SKXRC2025407).

References

[1] Kinan Dak Albab, Rawane Issa, Mayank Varia, and Kalman Graffi. 2022. Batched differentially private information retrieval. In *USENIX Security*.
[2] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: Differential privacy for location-based systems. In *CCS*. ACM.
[3] Apple and Google. 2021. Exposure Notification Privacy-preserving Analytics white paper. (2021). https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ENPA_White_Paper.pdf.
[4] Claudio A Ardagna, Marco Cremonini, Ernesto Damiani, S De Capitani di Vimercati, and Pierangela Samarati. 2007. Location privacy protection through obfuscation-based techniques. In *IFIP annual conference on data and applications security and privacy*. Springer, 47–60.
[5] Claudio A Ardagna, Marco Cremonini, Sabrina De Capitani di Vimercati, and Pierangela Samarati. 2009. An obfuscation-based approach for protecting location privacy. *IEEE Transactions on Dependable and Secure Computing* 8, 1 (2009), 13–27.
[6] Hilal Asi, Fabian Boemer, Nicholas Genise, Muhammad Haris Mughees, Tabitha Ogilvie, Rehan Rishi, Kunal Talwar, Karl Tarbe, Akshay Wadia, Ruiyu Zhu, et al. 2024. Scalable private search with wally. *arXiv preprint arXiv:2406.06761* (2024).
[7] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. 2019. The privacy blanket of the shuffle model. In *CRYPTO*. Springer.
[8] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. 2017. Prochlo: Strong privacy for analytics in the crowd. In *SOSP*.
[9] Junming Chen, Xiaoyue Ma, Lannan Luo, and Qiang Zeng. 2025. Tracking You from a Thousand Miles Away! Turning a Bluetooth Device into an Apple {AirTag} Without Root Privileges. In *USENIX Security*.

[10] Rui Chen, Haoran Li, A Kai Qin, Shiva Prasad Kasiviswanathan, and Hongxia Jin. 2016. Private spatial data aggregation in the local setting. In *IEEE ICDE*.

[11] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. 2019. Distributed differential privacy via shuffling. In *Eurocrypt*. Springer.

[12] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *SIGKDD*. ACM.

[13] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. 1998. Private information retrieval. *Journal of the ACM (JACM)* 45, 6 (1998), 965–981.

[14] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. 2012. Differentially private spatial decompositions. In *ICDE*. IEEE.

[15] Marc Damie, Florian Hahn, Andreas Peter, and Jan Ramon. 2025. How to Securely Shuffle? A survey about Secure Shufflers for privacy-preserving computations. *arXiv preprint arXiv:2507.01487* (2025).

[16] Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. 2018. Privacy pass: Bypassing internet challenges anonymously. *Proceedings on Privacy Enhancing Technologies* (2018).

[17] Karel Dhondt, Victor Le Pochat, Alexios Voulimeneas, Wouter Joosen, and Stijn Volckaert. 2022. A Run a Day Won't Keep the Hacker Away: Inference Attacks on Endpoint Privacy Zones in Fitness Tracking Social Networks. In *ACM CCS*.

[18] John C Duchi, Michael I Jordan, and Martin J Wainwright. 2018. Minimax optimal procedures for locally private estimation. *J. Amer. Statist. Assoc.* 113, 521 (2018), 182–201.

[19] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[20] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. 2019. Amplification by shuffling: From local to central differential privacy via anonymity. In *SODA*. SIAM.

[21] Vitaly Feldman, Audra McMillan, and Kunal Talwar. 2021. Hiding Among the Clones: A Simple and Nearly Optimal Analysis of Privacy Amplification by Shuffling. In *FOCS*. IEEE.

[22] Vitaly Feldman, Audra McMillan, and Kunal Talwar. 2022. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *IEEE FOCS*. IEEE.

[23] Vitaly Feldman, Audra McMillan, and Kunal Talwar. 2023. Stronger privacy amplification by shuffling for Rényi and approximate differential privacy. In *ACM-SIAM SODA*.

[24] Raghu K Ganti, Fan Ye, and Hui Lei. 2011. Mobile crowdsensing: current state and future challenges. *IEEE communications Magazine* 49, 11 (2011), 32–39.

[25] Adria Gascón, Yuval Ishai, Mahimna Kelkar, Baiyu Li, Yiping Ma, and Mariana Raykova. 2024. Computationally Secure Aggregation and Private Information Retrieval in the Shuffle Model. *Cryptology ePrint Archive* (2024).

[26] Bugra Gedik and Ling Liu. 2005. Location privacy in mobile systems: A personalized anonymization model. In *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. IEEE, 620–629.

[27] Bugra Gedik and Ling Liu. 2008. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing* 7, 1 (2008), 1–18.

[28] Badih Ghazi, Noah Golowich, Ravi Kumar, Rasmus Pagh, and Ameya Velingker. 2019. On the power of multiple anonymous messages. *arXiv preprint arXiv:1908.11358* (2019).

[29] Marco Gruteser and Dirk Grunwald. 2003. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*. 31–42.

[30] Mehmet Emre Gursoy, Ling Liu, Stacey Truex, Lei Yu, and Wenqi Wei. 2018. Utility-aware synthesis of differentially private and attack-resilient location traces. In *ACM CCS*.

[31] Apple Inc. n.d.. Location Services & Privacy. https://www.apple.com/legal/privacy/data/en/apple-maps/. [Accessed 17-04-2026].

[32] Google Inc. n.d.. How Google uses location information. https://policies.google.com/technologies/location-data. [Accessed 17-04-2026].

[33] Huawei Inc. n.d.. Privacy Policy, HUAWEI UK. https://consumer.huawei.com/uk/privacy/privacy-statement-huawei/. [Accessed 17-04-2026].

[34] Yuval Ishai, Mahimna Kelkar, Daniel Lee, and Yiping Ma. 2024. Information-Theoretic Single-Server PIR in the Shuffle Model. *Cryptology ePrint Archive* (2024).

[35] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. 2006. Cryptography from anonymity. In *FOCS*. IEEE.

[36] Zach Jorgensen, Ting Yu, and Graham Cormode. 2015. Conservative or liberal? Personalized differential privacy. In *2015 IEEE 31St international conference on data engineering*. IEEE, 1023–1034.

[37] Iris A Junglas and Richard T Watson. 2008. Location-based services. *Commun. ACM* 51, 3 (2008), 65–69.

[38] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2015. The composition theorem for differential privacy. In *ICML*. PMLR.

[39] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM J. Comput.* (2011).

[40] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. 2005. Protection of location privacy using dummies for location-based services. In *21st International conference on data engineering workshops (ICDEW'05)*. IEEE, 1248–1248.

[41] Jong Wook Kim and Beakcheol Jang. 2019. Workload-aware indoor positioning data collection via local differential privacy. *IEEE communications letters* 23, 8 (2019), 1352–1356.

[42] Antti Koskela, Joonas Jälkö, and Antti Honkela. 2020. Computing tight differential privacy guarantees using fft. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2560–2569.

[43] Xiaofeng Liu, Chaoshun Zuo, Qinsheng Hou, Pengcheng Ren, Jianliang Wu, Qingchuan Zhao, and Shanqing Guo. 2025. A thorough security analysis of BLE proximity tracking protocols. In *Proceedings of the 34th USENIX Conference on Security Symposium (SEC '25)*.

[44] Igal Sason and Sergio Verdú. 2016. f-divergence Inequalities. *IEEE Transactions on Information Theory* 62, 11 (2016), 5973–6006.

[45] Pengcheng Su, Haibo Cheng, and Ping Wang. 2026. Decomposition-Based Optimal Bounds for Privacy Amplification via Shuffling. *IEEE Symposium on Security and Privacy (SP)* (2026).

[46] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems* 10, 05 (2002), 557–570.

[47] Shun Takagi and Seng Pei Liew. 2026. Analysis of Shuffling Beyond Pure Local Differential Privacy. *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)* (2026).

[48] Kunal Talwar, Shan Wang, Audra McMillan, Vitaly Feldman, Pansy Bansal, Bailey Basile, Aine Cahill, Yi Sheng Chan, Mike Chatzidakis, Junye Chen, et al. 2024. Samplable anonymous aggregation for private federated data analysis. In *ACM CCS*.

[49] Raphael R Toledo, George Danezis, and Ian Goldberg. 2016. Lower-Cost ε-Private Information Retrieval. *Proceedings on Privacy Enhancing Technologies* 4 (2016), 184–201.

[50] Han Wang, Hanbin Hong, Li Xiong, Zhan Qin, and Yuan Hong. 2022. L-SRR: Local Differential Privacy for Location-Based Services with Staircase Randomized Response. In *CCS*. ACM.

[51] Shaowei Wang, Changyu Dong, Xiangfu Song, Jin Li, Zhili Zhou, Di Wang, and Han Wu. 2025. Beyond Statistical Estimation: Differentially Private Individual Computation via Shuffling. In *USENIX Security*. https://arxiv.org/abs/2406.18145.

[52] Shaowei Wang, Liusheng Huang, Yiwen Nie, Xinyuan Zhang, Pengzhan Wang, Hongli Xu, and Wei Yang. 2019. Local differential private data aggregation for discrete distribution estimation. *IEEE Transactions on Parallel and Distributed Systems* 30, 9 (2019), 2046–2059.

[53] Shaowei Wang, Yiwen Nie, Pengzhan Wang, Hongli Xu, Wei Yang, and Liusheng Huang. 2017. Local private ordinal data distribution estimation. In *INFOCOM*. IEEE.

[54] Shaowei Wang, Yun Peng, Kongyang Chen, and Wei Yang. 2024. Optimal locally private data stream analytics. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE, 31–40.

[55] Shaowei Wang, Yun Peng, Jin Li, Zikai Wen, Zhipeng Li, Shiyu Yu, Di Wang, and Wei Yang. 2024. Privacy Amplification via Shuffling: Unified, Simplified, and Tightened. *Proceedings of the VLDB Endowment* 17, 8 (2024), 1870–1883.

[56] Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* (1965).

[57] Zongda Wu, Guiling Li, Shigen Shen, Xinze Lian, Enhong Chen, and Guandong Xu. 2021. Constructing dummy query sequences to protect location privacy and query privacy in location-based services. *World Wide Web* 24 (2021), 25–49.

[58] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2014), 129–142.

[59] Mao Ye, Peifeng Yin, and Wang-Chien Lee. 2010. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. 458–461.

[60] Tingfeng Yu, James Henderson, Alwen Tiu, and Thomas Haines. 2024. Security and Privacy Analysis of Samsung's Crowd-Sourced Bluetooth Location Tracking System. In *USENIX Security*. USENIX Association.

[61] Hui Zang and Jean Bolot. 2011. Anonymization of location data does not work: A large-scale measurement study. In *Proceedings of the 17th annual international conference on Mobile computing and networking*. 145–156.

[62] Yu Zheng, Hao Fu, Xing Xie, Wei-Ying Ma, and Quannan Li. 2011. *Geolife GPS trajectory dataset - User Guide* (geolife gps trajectories 1.1 ed.). https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/

[63] Mingxun Zhou, Giulia Fanti, and Elaine Shi. 2024. Conan: Distributed Proofs of Compliance for Anonymous Data Collection. In *ACM CCS*.

[64] Yuqing Zhu, Jinshuo Dong, and Yu-Xiang Wang. 2022. Optimal accounting of differential privacy via characteristic function. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 4782–4817.