Privacy-Preserving Aggregate Mobility Data Release: An Information-Theoretic Deep Reinforcement Learning Approach

Wenjing Zhang, Bo Jiang, Ming Li, *Senior Member, IEEE*, and Xiaodong Lin, *Fellow, IEEE*

***Abstract*—It is crucial to protect users' location traces against inference attacks on aggregate mobility data collected from multiple users in various real-world applications. Most of the existing works on aggregate mobility data are focusing on inference attacks rather than designing privacy-preserving release mechanisms, and a few differential private release mechanisms suffer from poor utility-privacy tradeoffs. In this paper, we propose optimal centralized privacy-preserving aggregate mobility data release mechanisms (PAMDRMs) that minimize the leakage from an information-theoretic perspective by releasing perturbed versions of the raw aggregate location. Specifically, we use mutual information to measure user-level and aggregate-level privacy leakage separately, and formulate leakage minimization problems under utility constraints. As directly solving the optimization problems incur exponential complexity w.r.t. users' trace length, we transform them into belief state Markov Decision Processes (MDPs), with a focus on the MDP formulation for the user-level privacy problem. We build reinforcement learning (RL) models and leverage the efficient Asynchronous Advantage Actor-Critic RL algorithm to derive the solutions to the MDPs as our optimal PAMDRMs. We compare them with two state-of-the-art privacy protection mechanisms PDPR [1] (context-aware local design) and DMLM [2] (context-free centralized design) in terms of mutual information leakage and adversary's attack success (evaluated by her expected estimation error and Jensen-Shannon Divergence-based error). Extensive experimental results on both synthetic and real-world datasets demonstrate that the user-level PAMDRM performs the best on both measures thanks to its context-aware property and centralized design. Even though the aggregate-level PAMDRM achieves better privacy-utility tradeoff than the other two, it does not always perform better than them on adversarial success, highlighting the necessity of considering privacy measures from different perspectives to avoid overestimating the level of privacy offered to users, concurred with the insight given in [3]. Lastly, we discuss an alternative, fully data-driven approach to derive the optimal PAMDRM by leveraging adversarial training on limited data samples.**

***Index Terms*—Privacy metrics, aggregate mobility data, information-theoretic metrics, reinforcement learning approach.**

I. Introduction

The pervasive use of mobile phones and widespread deployment of sensors have led to a sharp increase of the amount of mobility data, such as location time-series or location traces. Analyzing mobility data patterns is indispensable in improving public transportation and health, especially in fighting COVID-19 pandemic [4]–[6], which contributes to better well-being of our society. For example, applications including navigation, signal control, traffic predicting, and managing traveler information rely on the analysis of mobility data [7]. Nevertheless, mobility patterns can reveal personal information such as home or workplace, hospital visits, lifestyles or even religion [8], [9], which are private and sensitive information that people hesitate to share publicly. Hence directly using individual mobility data is not recommended due to privacy concerns.

To avoid exposure of individual mobility data, location traces collected from multiple mobile devices are often aggregated in space and time, and then the aggregate result is released to the public. The aggregate mobility data represents an estimation of population-level mobility [10], which has been shown to be beneficial for numerous real-world applications. For instance, it is useful in monitoring how social distancing is followed to understand its effectiveness in the control of the COVID-19 pandemic [4]. Most recently, the aggregate data shared from Deutsche Telekom are effectively utilized for measuring social distancing [5]. To aid infection prevention from the pandemic, Apple also shares aggregate navigation data from Apple Maps to help detect mobility trends in different cities and countries [6]. In addition, aggregate mobility data are also broadly used in traffic model study [11], city planning [12], smart metering [13] and business [14].

However, aggregation cannot guarantee individual user's mobility privacy [15]–[17]. In particular, Xu et al. [15] reconstruct target users' location traces from aggregate mobility data with no prior information provided. Pyrgelis et al. [16] show that an adversary can leverage the aggregate location time-series to infer users' movements accurately. Besides, Membership Inference Attack (MIA) was successfully performed in [17] by assessing if a specific user contributes to the aggregates or not. Obviously, MIAs on aggregate mobility data violate user's privacy especially when the aggregate data connects with a group having a common sensitive characteristic, such as disease, religion, income, etc. Once an adversary identifies that someone has participated in a certain aggregated dataset, it can further deduce users' mobility profiles [16] or their location traces [15] accordingly.

To tackle the above privacy issue, a privacy-preserving aggregate mobility data release mechanism must be developed, which is typically done by generating noisy aggregates by perturbing the raw aggregates to a certain extent to pre-

Wenjing Zhang and Xiaodong Lin are with the School of Computer Science, University of Guelph, Guelph, Ontario, N1G2W1, Cananda. Email: *wzhang25@uoguelph.ca*; *xlin08@uoguelph.ca*.

Bo Jiang and Ming Li are with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, 85721. Email: *b-jiang@email.arizona.edu*; *lim@email.arizona.edu*.

Corresponding authors: Xiaodong Lin and Ming Li.

serve privacy, and releases the noisy version of aggregates instead. Unfortunately, most works to date focus on attacks on aggregate mobility data rather than protection mechanisms. Notwithstanding few existing mechanisms proposed based on Differential Privacy (DP) notion, they assume the worst-case adversary model that leads to worse utility-privacy tradeoff compared with context-aware notions (e.g., information-theoretic metrics). Accordingly, to design privacy-preserving aggregate mobility data release mechanisms that achieve better utility-privacy tradeoff, we aim to incorporate context (i.e., prior knowledge) to enhance utility by adding noise according to the prior, using an information-theoretic privacy metric with average sense privacy guarantee.

A. Related work

1) DP-based aggregate data release mechanisms: As discussed above, the works in [15]–[17] focus on inference algorithms or attacks on specific aggregate datasets rather than deriving explicit privacy-preserving release mechanisms. DP is a rigorous privacy notion based on indistinguishability, which is commonly used in protecting individual's data while releasing aggregate information about a database [18]–[22]. However, as a context-free metric, DP assumes the strongest adversary model, where it may have any background information and know any other users' data but the target's. As a result, DP incurs worse utility-privacy tradeoff compared with context-aware notions, such as information-theoretic metrics, as shown in [23]–[25]. Moreover, Pyrgelis *et al.* have shown that DP-based mechanisms are still vulnerable to inference attacks because they only guarantee indistinguishability but do not prevent concrete inference attacks [16]. These findings motivate us to adopt the information-theoretic metric that quantifies the actual leakage and can prevent data inference. In addition, Li *et al.* proposed a scheme using additive homomorphic encryption to efficiently achieve privacy-preserving aggregation [26]. Nevertheless, cryptographic methods alone cannot guarantee DP since it still reveals the accurate raw aggregates to the public, thus still leaks privacy.

2) Context-aware privacy metric and mechanisms: Designing privacy-preserving data release mechanisms based on context-aware privacy metrics [23], [27], [28] commonly refers to the case where data's prior knowledge [1] is considered in the privacy definition and exploited in mechanism design through adding selective noise according to the data priors, with an advantage of ensuring higher utility-privacy tradeoffs than context-unaware approaches [31]. Commonly used context-aware privacy metrics include attacker's average estimation error [32]–[34] or information-theoretic (IT) notions [1], [3], [25], [35]–[38]. As a context-aware metric, mutual information (MI) has operational meaning regarding capturing the expected information gain of the adversary about users' location data after observing the released aggregate output [31]. It is independent of the actual inference attack algorithm that adversary uses and her computational power. As a newly proposed stronger IT notion, maximal information leakage [37], [38] measures privacy in terms of adversary's gain in guessing the private information after observing released data. However, as discussed in [37], it fails to capture the importance of protecting highly-confidential data if those data can only take a few possible values, which can possibly happen on location data, so it may not be suitable for evaluating location privacy.

Even though MI has the advantage of ensuring higher privacy-utility tradeoff than other context-aware metrics, we should combine MI with different privacy metrics, such as the adversary's average estimation error [34] or Jensen-Shannon Divergence-based error [16] to add a new dimension to mutual information, as recommended by Oya et al. in a recent influential work [3]. This is because optimal privacy-preserving data release mechanisms derived based solely on one metric without being guided by a complementary metric can result in providing little utility or little privacy [3]. For instance, an optimal mechanism in terms of adversary's estimated error [34] suffers from near-zero privacy for users since their released locations have little uncertainty for the adversary, resulting in a small conditional entropy of locations. However, evaluating a mechanism using conditional entropy may give us a false perception of privacy, e.g., a mechanism having a small value of conditional entropy does not mean it can only provide a low privacy protection, because the entropy of the location prior is already low [3]. Therefore, MI must be considered to get a full picture of the information-theoretic leakage of the mechanism. Nonetheless, note that MI does not consider geographic distance (i.e., it do not capture the adversary's ability to estimate the real location) [3], we shall not evaluate a mechanism based solely on MI due to the geographic nature of our aggregate location privacy problem. Thus, we also utilize other geographical metrics (including attacker's average estimation error) to provide another perspective on quantifying the actual privacy leakage of our schemes in addition to mutual information leakage.

There are a few works in the literature that leverage mutual information as the privacy metric to design privacy-preserving location release mechanisms [1], [3], [25]. Specifically, the Blahut-Arimoto algorithm was adopted in [3] (individual user's sporadic location) and modified in [25] (individual user's location trace) to minimize the mutual information leakage on individual user's original sporadic location/location trace introduced from releasing perturbed location/trace. However, the works in [3], [25] do not consider history of all the release data in the past, resulting in sub-optimal leakage, because considering all history will result in exponential complexity w.r.t. trace length. It is worth noting that the work studied in [1] addresses this challenge by casting the original optimization problem as an MDP process, which can be efficiently solved via an RL approach. Nevertheless, there are two ways of extending, namely that the first one is a local mechanism for each user, the other one is centralized mechanism where we should take as input all the user's data at same time.

[1] Prior information is naturally available in the real world, which can be obtained from historical data or some similar/related published data. For example, people's location priors are sometimes similar especially in popular tourist destinations since they are expected to visit favored locations more frequently than normal ones [29]; as a useful reference in medical diagnosis, prior information like the probability of having a certain type disease is commonly accessible from the medical findings published before [30].

3) MDP and RL approaches in data privacy: Reinforcement learning (RL) is emerging as an efficient technique aiming to solve complex decision-making problems that can be formulated as MDPs, used as mathematical framework to describe RL models. Simon et al. [39] study the optimal battery charging policy for smart metering systems that minimizes information-theoretic privacy leakage subject to causality and charge conservation. Their main contribution is to propose a series of reductions on the original minimization problem and ultimately recast it as an MDP. Nevertheless, there is no concrete algorithm proposed to solve the MDP. Inspired by this work, Erdemir et al. [1] explored how to efficiently derive the optimal privacy-preserving individual location trace sharing mechanism using a deep RL approach, but its extension to aggregate mobility data from multiple users results in high adversarial success as shown in Section VI. We highlight that our work improves [1] in terms of mutual information leakage and adversarial success only if we apply [1] to a multi-user scenario, and the work in [1] is still optimal in protecting single user's location trace. Additionally, our problem poses a mathematical challenge in the MDP formulation for minimizing user-level leakage since the cost function captures an individual user's location, while the policy involves multiple users' aggregate locations. This brings the difficulty in expressing the cost and state update rule as functions of policy, although it is generally straightforward in a traditional MDP formulation.

B. Contributions

We formulate two optimization problems of deriving the optimal centralized privacy-preserving aggregate mobility data release mechanisms (shortened as PAMDRM), targeting on protecting the worst-case user-level privacy (PAMDRM_user) and aggregate-level privacy (PAMDRM_agg) separately. Particularly, we minimize the mutual information leakage incurred on an individual user's trace and the original aggregate mobility data when releasing the noisy aggregate given utility constraints, with the main focus on deriving the optimal PAMDRM_user. Since computational complexity issue arises from directly solving the optimization problems to derive the optimal PAMDRMs, we formulate them as belief state MDPs that can be efficiently solved via an asynchronous RL approach, where PAMDRMs are captured by the action probabilities. The major contributions are summarized as follows:

- We find it difficult to properly define the belief state, action probability, and cost function when formulating the user-level optimization problem as an MDP, and turn to identify a simplified yet equivalent sequential optimization problem inspired by [1], [39], followed by proving its upper bound. Mathematical proofs demonstrate that the belief MDP formulation is more challenging than previous work [1], [39] due to the complex policy structure that captures multiple users' participations and the temporal correlations in their mobility traces. We adopt the solution of the MDP formulated from the upper bound as the PAMDRM_user to release noisy aggregate mobility data, providing the privacy guarantee that the user-level leakage is no larger than the upper bound. More importantly, we prove that the user-level leakage introduced by releasing the noisy aggregate data is actually upper bounded by the aggregate-level leakage.
- We build a reinforcement learning model to solve the proposed belief MDPs and implement an efficient Asynchronous Advantage Actor-Critic (A3C) RL algorithm, where we train the actor and critic networks to output the best action probabilities as the optimal PAMDRMs. We perform extensive experimental evaluations on both synthetic and real-world datasets to evaluate and compare the privacy protection capabilities provided by PAMDRM_user and PAMDRM_agg with two state-of-the-art privacy protection methods PDPR [1] and DMLM [2]. Specifically, we implement an extension of PDPR (named EoPDPR) to our aggregate data release setting since it is not directly comparable due to its local mechanism design. Experimental results show that PAMDRM_user performs the best in terms of privacy-utility tradeoff, while DMLM incurs the largest leakage, since PAMDRM_user, PAMDRM_agg, and EoPDPR are all based on context-aware metric and thereby resulting in better privacy-utility tradeoffs. In addition, the result also validates our theoretical proof that the user-level leakage is upper bounded by the aggregate-level leakage.
- To provide another perspective of privacy metric in addition to MI leakage, we also evaluate adversarial success by performing the Bayesian updating location inference attack [16] on released location aggregate. The adversarial success is assessed by calculating her attack error measured by the Jensen-Shannon Divergence [16] and expected estimation error [34]. Experimental results show that PAMDRM_user performs better than others regarding adversarial success thanks to its context-aware property and centralized mechanism design. Interestingly, comparison between EoPDPR and DMLM indicates when the number of participants in aggregation is smaller, EoPDPR has lower adversarial attack success than DMLM due to its advantage of using context-aware metrics outweighing the disadvantage of using a local mechanism, but DMLM performs better on more participants since centralized design typically requires less amount of noise than local design under the same privacy level. Even though PAMDRM_agg is better than EoPDPR and DMLM on privacy-utility tradeoff, it does not always perform better than them on adversarial success. This also suggests we consider privacy measures from a different perspective to avoid overestimating the level of privacy offered to users, concurred with the insight in [3]. It is important to note that PDPR [1] is still optimal in protecting single user's location trace, while our work provides better performance in a multi-user scenario. Lastly, we design a data-driven based scheme to derive the optimal PAMDRM by leveraging adversarial training on limited data samples, where the key idea is to use a novel MI estimator [40] that has been shown to be effective in MI minimization.

The rest of this paper is organized as follows. We present the problem statement in Section II. Section III describes the main results for the aggregate/individual privacy - aggregate utility tradeoff. We formulate the proposed problems as MDPs and build an RL model in Section IV, and propose an A3C-based algorithm for generating the optimal PAMDRMs in Section V. Section VI presents experimental results, followed by discussions on a data-driven approach and alternative formulations of our problem in Section VII. Finally, we conclude our work and present potential future work in Section VIII. Important symbols and notations are summarized in Table I.

II. Problem Statement

In this section, the problem setting is introduced first, followed by the formulation of our proposed problem.

A. Problem Setting

We study a problem setting where M independent users participate in a location aggregation process [2] and each user's location trace is temporally correlated, as illustrated in Fig.1. To ease discussion, we consider a trusted location data aggregator, who first collects location traces from those M users, where user-m's location trace is denoted by $\boldsymbol{U}_m$. Next, the aggregator calculates a specific type of aggregate mobility data based on users' traces, e.g., summation, mean, or density. In this work, we consider calculating the statistic of summation $\boldsymbol{O}$ over users' traces as the aggregate mobility data (or aggregate location traces), which is especially important in people's spatial patterns analysis, such as area hotspots detection. However, directly releasing the raw aggregates $\boldsymbol{O}$ to the public discloses users' sensitive information, as discussed in Section I, so the aggregator will release its perturbed/noisy version $\boldsymbol{R}$ instead to protect their privacy.

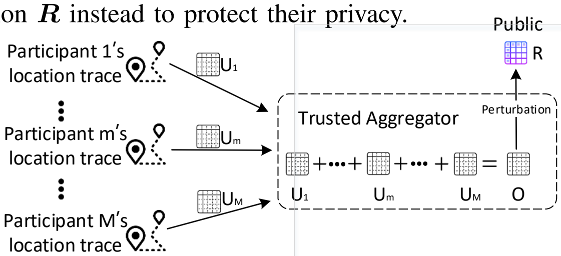
Public
R
Participant 1's location trace
U1
Participant m's location trace
Um
Participant M's location trace
UM
Trusted Aggregator
Perturbation
+•••+
+••• +
=
U1
Um
UM
O

Fig. 1. Privacy-preserving aggregate mobility data release via centralized perturbation mechanism.

Next, we formally define location traces and aggregate mobility data and then describe the threat model.

User's location trace. Each user's location trace is represented by a matrix $\boldsymbol{U}_m$ of size $L \times T$, with prior distribution $p(\boldsymbol{U}_m)$, where $m \in \{1, 2, ..., M\}$ denotes her index, L represents the total number of locations that users visit, and T is the total number of time steps in an aggregation process. To simplify our model, each user's trace length is truncated to size T. Each element $\boldsymbol{U}_m(l, t) = 1$ denotes that user-m is present at location $l \in \{1, 2, ..., L\}$ at time step $t \in \{1, 2, ..., T\}$ and 0 otherwise, i.e., each column vector in matrix $\boldsymbol{U}_m$ has one element as 1 and others as 0. We assume each user's trace follows the first-order Markov property, i.e., given her current location, the location at the next time step is independent of all the previous locations [25], [34], [45], [46].

TABLE I
Notations

Symbol	Description
t, T	Time step (integer), time period of aggregation
l, L	Location ID (integer), total number of locations
m, M	User ID (integer), total number of users participating in an aggregation process
U_m	User-m's location trace
O, R	Original, noisy location aggregates
$q(\cdot \mid \cdot)$, $p(\cdot, \cdot)$	Conditional, joint probability distributions
$a_t, b_t, \mathcal{C}_t, \mathcal{R}_t$	Action, belief state, cost, and reward function at time t
$\theta_\pi, \theta_v (\theta'_\pi, \theta'_v)$	Parameters of global (local) actor and critic networks
γ, η	Discount factor, learning rate

Aggregate mobility data. The aggregate mobility data generated by a trusted aggregator is denoted by a matrix $\boldsymbol{O}$ of size $L \times T$. In this paper, we are interested in studying the aggregate statistic of summation, thus we have $\boldsymbol{O} = \sum_{m=1}^{M} \boldsymbol{U}_m$. Each element $\boldsymbol{O}(l, t) \in \{1, 2, ..., M\}$ represents the total number of users who visit location l at time step t, and is calculated as $\boldsymbol{O}(l, t) = \sum_{m=1}^{M} \boldsymbol{U}_m(l, t)$. We consider the case where each user is present at only one location in the set of $\{1, 2, ..., L\}$ at any time t, so the sum of any column vector in matrix $\boldsymbol{O}$ equals to the total number of users, i.e., $\sum_{l=1}^{L} \boldsymbol{O}(l, t) = M$.

Threat model. The adversary (e.g., an untrusted service provider) under consideration has full access to the statistical knowledge of users' location priors and the raw aggregates, denoted by $p(\boldsymbol{U}_m)$ and $p(\boldsymbol{O})$ respectively, which is illustrated in Fig.2. Her goal is to make inference about a target user's location trace $\boldsymbol{U}_m$ and also the raw aggregates $\boldsymbol{O}$ once she observes the perturbed aggregates $\boldsymbol{R}$. We do not restrict her computational capability, so theoretically she is capable of leveraging her own knowledge and the perturbed aggregates $\boldsymbol{R}$ to perform any type of inference attack. By examining this threat model, we aim to understand the fundamental information/privacy leakage on any individual user-m's location trace $\boldsymbol{U}_m$ as well as the raw aggregates $\boldsymbol{O}$ introduced by releasing $\boldsymbol{R}$ from an information-theoretic point of view.

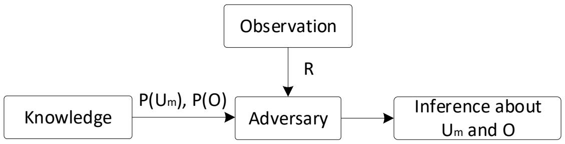

Fig. 2. Adversary model.

Metrics. We define privacy and utility metrics for measuring the privacy leakage and distortion in our problem. We replace the notations $\boldsymbol{U}_m$, $\boldsymbol{O}$, and $\boldsymbol{R}$ with $U_m^T = (U_1, ..., U_T)$, $O^T = (O_1, ..., O_T)$, and $R^T = (R_1, ..., R_T)$ to ease presentation.

***Definition** 1:* **Privacy Metrics.** *The information leakage occurring on the raw aggregate mobility data and an individual user-m's location trace introduced by releasing the*

[2] In real-life applications, the aggregation can be performed by a trusted aggregator [16], [22], [41] (such as a publisher who is bound by contractual obligations or a government organization) or using cryptographic protocols to blind individual locations when the server is untrusted to achieve private aggregation [42]–[44].

perturbed aggregates are measured by the mutual information between O^T and R^T, and the mutual information between U_m^T and R^T respectively, denoted by $I(O^T;R^T)$ and $I(U_m^T;R^T)$.

Intuitively, the more the raw aggregates are perturbed, the less privacy leakage will incur. However, the extent of perturbation should be limited to a certain amount so as to preserve data utility, so we define the following utility metric.

***Definition** 2:* **Utility Metric.** *The utility metric for aggregate mobility data is defined as $D(O^T;R^T) = \sum_{t=1}^{T} D(O_t,R_t)$, where $D(O_t,R_t)$ denotes the expected distortion at time t, and is defined as $D(O_t,R_t) = \sum_{o_t,r_t} p(o_t,r_t)d(o_t,r_t)$, and $d(O_t,R_t)$ is the Euclidean norm (i.e., L2 norm) of vector $O_t - R_t$. The utility constraint at time t is defined as $D(O_t,R_t) \leq D_t, t = 1,2,...,T$, implying that the total distortion for aggregate mobility data is $D \leq \sum_{t=1}^{T} D_t$.*

B. Problem formulation

We define the following Aggregate and Individual Privacy – Aggregate Utility tradeoff separately to formulate the problems of minimizing the aggregate-level and user-level privacy leakage subject to distortion constraints.

***Definition** 3:* **Aggregate Privacy – Aggregate Utility tradeoff (Aggregate-level Leakage):** *In an online aggregate mobility data release setting, given the utility constraint D_t at time step t, the tradeoff is defined as*

$$\boldsymbol{P1}: \quad \mathcal{L}_{agg}^{*}(D) = \min_{\substack{q_t(r_t|o^t,r^{t-1}): \\ \{D(O_t,R_t)\leq D_t\}_{t=1}^{T}}} I^q(O^T;R^T),$$

where $q_t(r_t|o^t,r^{t-1})$ represents the PAMDRM_agg at time step t, i.e., the current perturbed aggregates r_t is sampled from a conditional probability distribution $q_t(r_t|o^t,r^{t-1})$ given all the raw aggregates o^t and the release history r^{t-1}. Here we use o^t and r^{t-1} to denote $(o_1,o_2,...,o_t)$ and $(r_1,r_2,...,r_{t-1})$ respectively, and r_t has the same alphabet as o_t, i.e., r_t takes values from the set of realizations of o_t.

***Definition** 4:* **Individual Privacy – Aggregate Utility tradeoff (Worst-Case User-level Leakage):** *In an online aggregate mobility data release setting, given the utility constraint D_t at time step t, the tradeoff is defined as*

$$\boldsymbol{P2}: \quad \mathcal{L}_{user}^{*}(D) = \min_{\substack{q_t(r_t|o^t,r^{t-1}): \\ \{D(O_t,R_t)\leq D_t\}_{t=1}^{T}}} \max_{1\leq m\leq M} I^q(U_m^T;R^T).$$

This tradeoff can be interpreted as finding the optimal PAMDRM_user $q_t(r_t|o^t,r^{t-1})$ at each time step t that minimizes the maximal mutual information between U_m^T and R^T among all users under a distortion constraint D_t. However, directly solving $\boldsymbol{P1}$ and $\boldsymbol{P2}$ will induce exponential complexity.

Computational challenge. The mutual information in the objective functions in $\boldsymbol{P1}$ and $\boldsymbol{P2}$ are calculated as $I(O^T;R^T) = \sum_{o^T,r^T} p(o^T,r^T)\log\frac{p(o^T,r^T)}{p(o^T)p(r^T)}$ and $I(U_m^T;R^T) = \sum_{u_m^T,r^T} p(u_m^T,r^T)\log\frac{p(u_m^T,r^T)}{p(u_m^T)p(r^T)}$, which are difficult since they are mutual information expressions over history dependent probability distributions of random matrices O^t, R^t, and U_m^t. Specifically, calculation of $I(O^T;R^T)$ and $I(U_m^T;R^T)$ involve $|O^t||R^t|$ and $|U_m^t||R^t|$ operations that grow exponentially with the increase in trace length T. Motivated by [1], [39], we transform $\boldsymbol{P1}$ and $\boldsymbol{P2}$ into sequential optimization problems that can be formulated as belief state MDPs with policies capturing the optimal release mechanisms and then adopt an RL approach so as to address this issue.

III. Main Results

In this section, we prove the main theoretical results for Aggregate and Individual Privacy – Aggregate Utility tradeoffs separately, with the focus on the highlight of the latter. These results are the mathematical foundations that are key to further MDP formulation, which allows us to avoid exponential computation complexity in directly solving $\boldsymbol{P1}$ and $\boldsymbol{P2}$.

A. Result for Aggregate Privacy – Aggregate Utility Tradeoff

Note that an obstacle in obtaining an RL decomposition for $\boldsymbol{P1}$ is that the objective function is not of the form $\sum_{t=1}^{T} cost(state_t, action_t)$. Fortunately, the inspirational work from [1], [39] already paved the way for addressing this difficulty. In particular, the formulation of $\boldsymbol{P1}$ is similar to the optimization problem in Eq.(2) in [1], even though we are dealing with the data structure with higher dimension in the form of a matrix rather than a vector. Essentially, we are able to follow their idea to prove the following theorem, and the details of proof are presented in Appendix A of the supplementary document.

We start with defining the simplified set of mechanisms. Let Q_A denote the set of release mechanism $q_t^a(r_t|o^t,r^{t-1})$ in $\boldsymbol{P1}$, and $Q_B \in Q_A$ denote the set of release mechanisms that choose the perturbed aggregates r_t only conditioning on the current and previous raw aggregates o_t, o_{t-1}, and the released history r^{t-1}. That is, for any $q \in Q_B$, the perturbed aggregates r_t is r with probability $q_t^b(r|o_t,o_{t-1},r^{t-1})$. In particular, we define the release mechanism for the first time step as $q_1^a(r_1|o_1) = q_1^b(r_1|o_1)$.

***Theorem** 1: There is no loss of optimality in $\boldsymbol{P1}$ by restricting the original mechanism $q_t^a(r_t|o^t,r^{t-1}) \in Q_A$ to the simplified mechanism $q_t^b(r|o_t,o_{t-1},r^{t-1}) \in Q_B$. In addition, for any $q_t^b(r|o_t,o_{t-1},r^{t-1}) \in Q_B$, $\boldsymbol{P1}$ is equivalent to*

$$\boldsymbol{P3}: \quad \mathcal{L}_{agg}^{*}(D) = \min_{\substack{q_t^b(r_t|o_t,o_{t-1},r^{t-1}): \\ \{D(O_t,R_t)\leq D_t\}_{t=1}^{T}}} \sum_{t=1}^{T} I(O_t,O_{t-1};R_t|R^{t-1}),$$

where

$$I(O_t,O_{t-1};R_t|R^{t-1}) = \sum_{o_t,o_{t-1},r^t} p(o_t,o_{t-1},r^t)\log\frac{q_t^b(r_t|o_t,o_{t-1},r^{t-1})}{p(r_t|r^{t-1})}. \quad (1)$$

Theorem 1 indicates that simplifying release mechanism in terms of trace length can still preserve optimality in $\boldsymbol{P1}$ and the objective function can be written in an additive form.

B. Result for Individual Privacy – Aggregate Utility Tradeoff

A straightforward method for solving $\boldsymbol{P2}$ is summarized as follows. Firstly, we enumerate on all possible $q_t(r_t|o^t,r^{t-1})$ that satisfies a given utility constraint, and for each $q_t(r_t|o^t, r^{t-1})$ we calculate $I_m^q(U_m^T; R^T)$ for each user-m and save the maximum value of mutual information; among all these values, we select the minimum and the corresponding $q_t(r_t|o^t, r^{t-1})$ that yields this minimum value as the optimal release mechanism. However, this process involves enumeration over all possible $q_t(r_t|o^t, r^{t-1})$, which is impractical since the value of $q_t(r_t|o^t, r^{t-1})$ is taken from a continuous space. Fortunately, the following main results show that $\boldsymbol{P2}$ is actually equivalent to a new optimization problem $\boldsymbol{P4}$, with a provable upper bound $\boldsymbol{P5}$ that can be further formulated as an belief state MDP, where an efficient RL approach can be adopted to solve this MDP. The theorem below shows that the same simplification on the original release mechanism does not yield any loss of the optimality in $\boldsymbol{P2}$ and its objective function can also be written in an additive form.

***Theorem** 2: There is no loss of optimality in $\boldsymbol{P2}$ by restricting the original mechanism $q_t^a(r_t|o^t, r^{t-1}) \in Q_A$ to the simplified mechanism $q_t^b(r|o_t, o_{t-1}, r^{t-1}) \in Q_B$. In addition, for any $q_t^b(r|o_t, o_{t-1}, r^{t-1}) \in Q_B$, $\boldsymbol{P2}$ is equivalent to*

$$\boldsymbol{P4}: \mathcal{L}^*_{user}(D) = \min_{\substack{q_t^b(r_t|o_t,o_{t-1},r^{t-1}):\\ \{D(O_t,R_t)\le D_t\}_{t=1}^T}} \max_{1\le m\le M} \sum_{t=1}^{T} I_m(U_t, U_{t-1}; R_t|R^{t-1}),$$

which is upper bounded by

$$\boldsymbol{P5}: \mathcal{L}^*_{user}(D) = \min_{\substack{q_t^b(r_t|o_t,o_{t-1},r^{t-1}):\\ \{D(O_t,R_t)\le D_t\}_{t=1}^T}} \sum_{t=1}^{T} \max_{1\le m\le M} I_m(U_t, U_{t-1}; R_t|R^{t-1}),$$

where $I_m(U_t, U_{t-1}; R_t|R^{t-1})$ denotes the mutual information that corresponds to the locations U_t, U_{t-1} taking from user-m.

Proving Theorem 2 depends on the following two lemmas.

***Lemma** 1:* For any $q \in Q_A$,

$$\max_{1\le m\le M} I_m^q(U^T; R^T) \ge \max_{1\le m\le M} \sum_{t=1}^{T} I_m^q(U_t, U_{t-1}; R_t|R^{t-1})$$

with equality if and only if $q \in Q_B$.

***Lemma** 2: For any q_a, there exists a q_b such that*

$$\max_{1\le m\le M} \sum_{t=1}^{T} I_m^{q_a}(U_t, U_{t-1}; R_t|R^{t-1}) = \max_{1\le m\le M} \sum_{t=1}^{T} I_m^{q_b}(U_t, U_{t-1}; R_t|R^{t-1}),$$

where $q_a = (q_1^a, ..., q_t^a, ..., q_T^a)$ and $q_b = (q_1^b, ..., q_t^b, ..., q_T^b)$. Here q_t^a and q_t^b represent the probability $q_t^a(r_t|o^t, r^{t-1})$ and $q_t^b(r_t|o_t, o_{t-1}, r^{t-1})$ respectively, and q_t^b is constructed by $q_t^b(r_t|o_t, o_{t-1}, r^{t-1}) = q_t^a(r_t|o_t, o_{t-1}, r^{t-1})$.

We prove Lemma 1 and 2 in Appendix B and C of the supplementary document. Briefly speaking, the proofs show that for any q_a, there exists a q_b such that $I^{q_a}(U^T; R^T) \ge I^{q_b}(U^T; R^T)$, meaning that it does not yield any loss of optimality in restricting the release mechanisms to the form of q_b. In addition, for any q_b, the objective function can be rewritten as an additive form $\max_{1\le m\le M} \sum_{t=1}^{T} I_m^{q_b}(U_t, U_{t-1}; R_t|R^{t-1})$. Hence the proof of Theorem 2 is completed.

Theorem 1 and 2 indicate that observing r_t reveals information only about (o_t, o_{t-1}) rather than the entire history o^t by releasing noisy aggregates following $q_t(r_t|o_t, o_{t-1}, r^{t-1})$ at each time step. More importantly, by using $q_t(r_t|o_t, o_{t-1}, r^{t-1})$ as the structural simplification for the original release mechanism $q_t(r_t|o^t, r^{t-1})$, computational complexity regarding trace length T will be reduced since o^t is replaced by o_t, o_{t-1} in the release mechanisms. Despite Theorem 2 might seem similar to the results in [1], [39] at first sight, its proof is actually more complicated than the proof for Theorem 1 since the objective function captures an individual user's location, while the decision involves multiple users' aggregated locations, bringing the difficulty in expressing the objective as a function of the decision.

C. Connection between $\mathcal{L}^*_{user}(D)$ and $\mathcal{L}^*_{agg}(D)$

Interestingly, there is also a connection between $\mathcal{L}^*_{agg}(D)$ in $\boldsymbol{P1}$ and $\mathcal{L}^*_{user}(D)$ in $\boldsymbol{P2}$, as shown in the following theorem.

***Theorem** 3: $\mathcal{L}^*_{user}(D)$ is upper bounded by $\mathcal{L}^*_{agg}(D)$:*

$$\mathcal{L}^*_{\text{user}}(D) \le \mathcal{L}_{\text{user}}(D, \text{PL}^*_{agg}) \le \mathcal{L}^*_{\text{agg}}(D),$$

$\mathcal{L}_{\text{user}}(D, \text{PL}^*_{agg}) = \max_{1\le m\le M} I(U_m^T; R^T)|_{q_t^u(r_t|o^t,r^{t-1})=\text{PL}^*_{agg}}$, and PL is shorted for policy.

We prove Theorem 3 in Appendix D of the supplementary document. It provides the privacy guarantee that releasing aggregate mobility data according to PL^*_{agg} introduces at most $\mathcal{L}_{\text{user}}(D, \text{PL}^*_{agg})$ bits of privacy leakage on any individual user.

***Remark** 1: Note that the above main results assume that users' location traces follow the first-order Markov property. However, as discussed in [1], [39], the theoretical proofs can naturally extend to more realistic scenarios where users' traces are actually higher-order Markovian. For example, let us consider a general case where users' traces follows n-th order Markov property. To derive similar results, we can define new processes $\{\hat{U}_t\}_{t\ge1}$ and $\{\hat{O}_t\}_{t\ge1}$ where $\hat{U}_t = (U_{t-n+1}, ..., U_t)$ and $\hat{O}_t = (O_{t-n+1}, ..., O_t)$, and replace U_t and O_t with $\hat{U}_t$ and $\hat{O}_t$ respectively in Theorem 1 and 2.*

IV. MDP Formulation

In this section, we present how to formulate $\boldsymbol{P3}$ and $\boldsymbol{P5}$, namely the sequential optimization problems for Aggregate and Individual Privacy – Aggregate Utility tradeoffs, as MDPs with belief states. Based on the MDP formulation, we show that the minimum information leakage and optimal release mechanisms can be obtained by utilizing a reinforcement learning approach, which is effective in solving MDPs with complex states. To facilitate the forthcoming discussion of how to design the RL model of our privacy-preserving aggregate mobility data release problem, we first briefly present the background of MDP and belief state MDP, where the latter is the key to the MDP formulation of our problem.

A. Background of MDP and Belief State MDP

An MDP is an environment where all states are Markov. It is represented by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where the elements represent a finite set of states, a finite set of actions, state transition probability matrix, reward function, and the discount factor, respectively. A policy $\pi(a|s) = p(A_t = a|S_t = s)$ is a distribution over actions given states. The state value function $V_\pi(s) = \mathbb{E}[G_t|S_t = s]$ of an MDP is the expected return starting from state s, and then following policy π, which evaluates the goodness of state s, where $G_t = \sum_{k=0}^{+\infty} \gamma^k R_{t+k+1}$ is the return. We consider an MDP is solved when we know the optimal state value function $V_*(s)$, i.e., the maximum value function over all policies. However, MDPs assume a scenario where the complete states of the world are visible to an agent, which is unrealistic because commonly these states cannot be directly observed in real world [47]. Unlike the problem of finding a mapping from states to actions in standard MDPs, the problem in Partially Observable Markov Decision Process (POMDP) is to find a mapping from a probability distribution (over states) to actions. In particular, the probability distribution over states is referred to as a belief state, and the entire probability space is referred to as the belief space [48]. The optimal solution to a POMDP gives the optimal action for every possible belief over states, which maximizes the expected reward of an agent, and the sequence of these optimal actions is the optimal policy. Simply put, a POMDP can be seen as a continuous space belief MDP, which can be solved by an RL approach. Interestingly, the belief state defined for our problem is proved to be Markovian, implying the possibility of being formulated as a belief state MDP.

B. MDP Formulation of $\boldsymbol{P3}$

1) A Sequential Optimization Problem:

***Proposition** 1: $\{O_t\}_{t\geq 1}$ is a first-order Markov chain when users are independent.*

The proof of Proposition 1 is shown in Appendix E of the supplementary document. The intuition is that all users' location traces follow the first-order Markov property and are independent of each other, and we have $O_t = \sum_{m=1}^{M} U_m(t)$, i.e., O_t is a function of $U_m(t)$. Therefore, given O_t, O_{t+1} is independent of $O_1, O_2, ..., O_{t-1}$, meaning that $\{O_t\}_{t\geq 1}$ follows a first-order Markov process.

***Proposition** 2: $\boldsymbol{P3}$ is a controlled Markov process where the state at time t is O_t, the observation is R_t, and the actions are probabilities taking continuous values from the conditional probability distribution $p_t(r_t|o_t, o_{t-1}, r^{t-1})$ that satisfies the distortion constraint $D(O_t, R_t) \leq D_t$.*

Note that the adversary can only partially observe the states, i.e., she is only able to observe R_t rather than O_t, and Proposition 1 has proved the raw aggregate sequence $\{O_t\}_{t\geq 1}$ is a Markov process. Accordingly, $\boldsymbol{P3}$ is formulated as a Markov process with hidden state O_t, observation R_t, per time step cost of $\mathcal{C}_t = I(O_t, O_{t-1}; R_t|R^{t-1})$, i.e., the information leakage at time step t, and action of $a_t^A(r_t|o_t, o_{t-1}) = q_t(r_t|o_t, o_{t-1}, r^{t-1}))$. After the agent takes action $a_t^A(r_t|o_t, o_{t-1})$, the current observation R_t is chosen according to the conditional probability $q_t(r_t|o_t, o_{t-1}, r^{t-1})$ and the state O_t evolves according to $p(o_t|o_{t-1})$. The objective is to find the optimal policy $q = (q_1, ..., q_T)$ to minimize the the total expected cost for the above sequential optimization problem. To do this, we construct an RL decomposition for the above Markov process.

2) A Reinforcement Learning Decomposition: The above Markov process described in Proposition 2 is slightly different from a POMDP. This is because the per time step cost in (1) involves the observation history r^{t-1}, contrary to the traditional cost model used in a POMDP, which only depends on the current action and state. However, this process can be formulated as a standard MDP if we consider the agent's belief as the state [39]. The rationale behind this idea is that the belief state MDP is not partially observable anymore since at any given time the agent knows its belief and by extension the state of the belief MDP. Now the policy is defined as a mapping from belief states to action probabilities. The belief state and action probability (i.e., PAMDRM) in our belief MDP are defined as follows. For any realization r^{t-1} of past observations, the belief state is defined as $b_t^A(o_{t-1}) = p(o_{t-1}|r^{t-1})$, namely the probability distribution over states conditioned on r^{t-1}, and the action induced by a release mechanism $q(r_t|o_t, o_{t-1}, r^{t-1})$ is defined as $a_t^A(r_t|o_t, o_{t-1}) = q_t(r_t|o_t, o_{t-1}, b_t^A(o_{t-1}))$. The belief state at each time is updated according to

$$\begin{aligned} b_{t+1}^A(o_t) = p(o_t|r^t) = \frac{\sum_{o_{t-1}} p(o_t, o_{t-1}, r^t|r^{t-1})}{\sum_{o_t, o_{t-1}} p(o_t, o_{t-1}, r_t|r^{t-1})} \\ = \frac{\sum_{o_{t-1}} p(o_t|o_{t-1}) q_t(r_t|o_t, o_{t-1}, r^{t-1}) p(o_{t-1}|r^{t-1})}{\sum_{o_t, o_{t-1}} p(o_t|o_{t-1}) q_t(r_t|o_t, o_{t-1}, r^{t-1}) p(o_{t-1}|r^{t-1})} \\ = \frac{\sum_{o_{t-1}} p(o_t|o_{t-1}) a_t^A(r_t|o_t, o_{t-1}) b_t^A(o_{t-1})}{\sum_{o_t, o_{t-1}} p(o_t|o_{t-1}) a_t^A(r_t|o_t, o_{t-1}) b_t^A(o_{t-1})}. \end{aligned} \tag{2}$$

Given the belief and action, the mutual information $I(O_t, O_{t-1}; R_t|R^{t-1})$ in $\boldsymbol{P3}$, i.e., the per time step cost sent to the agent is denoted by $C_t^A(b^A, a^A)$ and calculated as

$$\begin{aligned} & C_t^A(b_t^A, a_t^A) = I(O_t, O_{t-1}; R_t|R^{t-1}) \\ = & \sum_{o_t, o_{t-1}, r^t} p(o_t, o_{t-1}, r^t) \log \frac{q_t(r_t|o_t, o_{t-1}, r^{t-1})}{p(r_t|r^{t-1})} \\ = & \sum_{o_t, o_{t-1}, r^t} b_t^A(o_{t-1}) a_t^A(r_t|o_t, o_{t-1}) p(o_t|o_{t-1}) \\ & \times \log \frac{a_t^A(r_t|o_t, o_{t-1})}{\sum_{\hat{o}_t, \hat{o}_{t-1}} b_t^A(\hat{o}_{t-1}) a_t^A(r_t|\hat{o}_t, \hat{o}_{t-1}) p(\hat{o}_t|\hat{o}_{t-1})}. \end{aligned} \tag{3}$$

The objective of the agent is to minimize its long-term time-average expected cost $\mathcal{C}^A \triangleq \lim_{t\to\infty} \frac{1}{t} \mathbb{E}\{\sum_{\tau=1}^{t} C_\tau^A(b_\tau^A, a_\tau^A)\}$, under the distortion constraint $D(O_t, R_t) \leq D_t$ that is imposed at each time step t. We illustrate the belief state RL model in Fig. 3, where the environment is modeled as a finite state machine with inputs (actions sent from the agent) and outputs (observations and cost sent to the agent).

The agent represents the aggregator, and we set his knowledge to be the same as the adversary (who cannot observe the true states), so the leakage derived from the RL-based algorithm is exactly the leakage that occurred when the adversary observes the released noisy aggregates. This is also the reason for formulating our problem as a belief state MDP instead of an MDP since the agent cannot observe the true states. We finally formulate $\boldsymbol{P3}$ as a continuous state and action space MDP that can be solved by an RL approach.

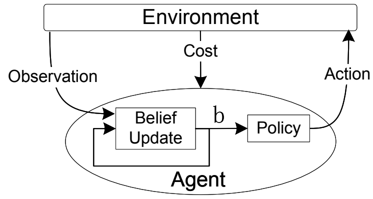

Fig. 3. RL model of privacy-preserving aggregate mobility data release.

C. MDP Formulation of $\boldsymbol{P5}$

Even though formulating $\boldsymbol{P5}$ for Individual Privacy – Aggregate Utility tradeoff as an MDP may follow a similar idea as above, deriving the corresponding belief state update function and the cost model is more challenging since the mutual information (i.e., objective function) in $\boldsymbol{P5}$ is difficult to connect with the release mechanism (i.e., decision policy). We present the following proposition as the first step towards the belief state MDP formulation of $\boldsymbol{P5}$.

***Proposition** 3: $\{O_t, U_t\}_{t\geq 1}$ is a first-order Markov chain when users are independent.*

The proof of Proposition 3 is given in Appendix F of the supplementary document. In light of the above explanation of the belief MDP formulation for $\boldsymbol{P3}$, we intend to formulate $\boldsymbol{P5}$ as a belief MDP as well. Specifically, based on Proposition 3, we define the belief state as $b_t^U(o_{t-1}, u_{t-1}) = p(o_{t-1}, u_{t-1}|r^{t-1})$, and prove that $\{b_t^U\}_{t\geq 1}$ is a controlled Markov process with control action $a_t^U(r_t|o_{t-1}, o_t) = p(r_t|o_{t-1}, o_t, r^{t-1})$. The following lemma shows the belief state update function, and its proof is given in Appendix G of the supplementary document.

***Lemma** 3: The belief state update function is*

$$b_{t+1}^U(o_t, u_t) = p(o_t, u_t|r^t) = \frac{\sum_{o_{t-1}} a_t^U(r_t|o_{t-1}, o_t) \sum_{u_{t-1}} p(o_t|o_{t-1})p(u_t|u_{t-1})b_t^U(o_{t-1}, u_{t-1})}{p(r_t|r^{t-1})}, \tag{4}$$

where

$$p(r_t|r^{t-1}) = \sum_{o_t, o_{t-1}} a_t^U(r_t|o_{t-1}^t)\cdot \frac{\sum_{u_{t-1}^t} p(o_t|o_{t-1})p(u_t|u_{t-1})b_t^U(o_{t-1}, u_{t-1})\mathbb{1}_{a_t^U}(p(r^{t-1}))}{\sum_{u_{t-1}^t, o_{t-1}^t} p(o_t|o_{t-1})p(u_t|u_{t-1})b_t^U(o_{t-1}, u_{t-1})\mathbb{1}_{a_t^U}(p(r^{t-1}))}, \tag{5}$$

and $\mathbb{1}_{a_t^U}(p(r^{t-1}))$ denotes an indicator function of the set that equals 1 if r^{t-1} is the observation that contributes to a_t^U and zero otherwise.

According to Lemma 3, the belief state at time $t+1$ is updated only depending on the previous belief state, the action at time t, and the transition probability distribution of $p(o_t|o_{t-1})$ and $p(u_t|u_{t-1})$, which can be readily calculated from a given dataset. Therefore, Lemma 3 implies that $\{b_t^U(o_{t-1}, u_{t-1})\}_{t\geq 1}$ is a controlled Markov process with control action $a_t^U(r_t|o_t, o_{t-1})$.

***Lemma** 4: The mutual information in $\boldsymbol{P5}$ is defined as the per time step cost, which can be written as a function of the belief state b_t and action a_t^U as*

$$I(U_t, U_{t-1}; R_t|R^{t-1}) = \sum_{u_t, u_{t-1}, r^t} p(u_t, u_{t-1}, r^t) \log \frac{p(r_t|u_t, u_{t-1}, r^{t-1})}{p(r_t|r^{t-1})}, \tag{6}$$

where

$$p(u_t, u_{t-1}, r^t) = \sum_{o_t, o_{t-1}} a_t^U(r_t|o_{t-1}^t)p(o_t|o_{t-1})p(u_t|u_{t-1})\cdot b_t^U(o_{t-1}, u_{t-1})\mathbb{1}_{a_t^U}(p(r^{t-1})),$$

$$p^{q_a}(r_t|u_t, u_{t-1}, r^{t-1}) = \sum_{o_t, o_{t-1}} a_t^U(r_t|o_{t-1}^t)\cdot \frac{p(o_t|o_{t-1})p(u_t|u_{t-1})b_t^U(o_{t-1}, u_{t-1})\mathbb{1}_{a_t^U}(p(r^{t-1}))}{\sum_{o_t, o_{t-1}} p(o_t|o_{t-1})p(u_t|u_{t-1})b_t^U(o_{t-1}, u_{t-1})\mathbb{1}_{a_t^U}(p(r^{t-1}))},$$

and $p(r_t|r^{t-1})$ is calculated according to (5).

The proof of Lemma 4 is presented in Appendix H of the supplementary document. We denote $\max_{1\leq m\leq M} I_m^q(U_t, U_{t-1}; R_t|R^{t-1})$ as C_t^U, i.e., the cost incurred at time t. The agent's objective is to minimize its long-term time-average expected cost $\mathcal{C}^U \triangleq \lim_{t\to\infty} \frac{1}{t}\mathbb{E}\{\sum_{\tau=1}^t C_\tau^U(b_\tau^U, a_\tau^U)\}$, under the distortion constraint $D(O_t, R_t) \leq D_t$ that is imposed at time step t.

Based on the above results, we can formulate $\boldsymbol{P5}$ as a continuous state and action space MDP. At each time step t, the agent receives a belief state b_t^U and selects an action a_t^U from the action set according to policy $\pi(a_t^U|b_t^U)$. In return, the agent receives the next state b_{t+1}^U and a scalar cost C_t^U. The process continues until the agent reaches a terminal state.

V. An Asynchronous RL approach

In this section, we explain the motivation for using an asynchronous RL approach and show how to make adaptations to solve the belief MDPs formulated from $\boldsymbol{P3}$ and $\boldsymbol{P5}$, representing the Aggregate and Individual Privacy – Aggregate Utility tradeoffs respectively.

A. Motivation for Using An A3C Approach

To avoid the complexity issue arising from the continuous states and actions in the proposed belief MDPs, we adopt policy gradient methods as the RL techniques, which optimize parameterized policies with respect to the long-term cumulative reward by gradient descent. However, vanilla policy gradients suffer from noticeable issues of noisy gradients and high variance, leading to instability and slow convergence [49]. As one class of policy gradient methods, Actor-Critic has the advantage of reducing variance and increasing stability. As a classic variant of the Actor-Critic method, Asynchronous Advantage Actor-Critic [50] (short for A3C) especially focuses on parallel training by using asynchronous gradient descent for optimization of deep neural network policies [50], where the critics learn the value function while multiple actors are trained in parallel and get synced with global parameters from time to time. Therefore, it ensures faster convergence surpassing the state-of-the-art Actor-Critic variants. It can be used in discrete as well as continuous action spaces.

B. Training Actor and Critic Networks in A3C Algorithm

A standard A3C framework contains a global network and several worker agents [3]. The global network interacts with the workers asynchronously, and each of them has a copy of the network and interacts with its own environment independently. Once an episode is finished, the global network receives the accumulated gradients, and then sends the updated parameters to the workers. Each worker agent maintains an estimated policy $\pi(a_t|b_t;\theta_\pi)$ [4] and an estimated value function $V_\pi(b_t;\theta_v)$, where θ_π and θ_v are the parameters of the actor and critic networks respectively, and learned in training stages.

Algorithm 1: Generating Privacy-Preserving AMDRM using A3C- Pseudocode for each actor worker thread

```
Input:
    • Belief state $b_t$; number of workers (CPU threads): $W$
    • Parameters of global (local) actor and critic networks: $\theta_\pi$, $\theta_v$ ($\theta'_\pi$, $\theta'_v$); Discount factor: $\gamma$; learning rate: $\eta$; exploration rate: $\epsilon$
    • Initialize global counter $T = 0$ and local counter $t = 1$
    • Maximum number of global shared counter: $T_{max}$
    • Maximum number of thread step counter: $t_{max}$
Output: Reward $r$; policy $\pi(a \mid b;\theta_\pi)$
 1: while $T < T_{max}$ do
 2:     for $w = 1$ to $W$ do
 3:         Reset global gradient $d\theta_\pi \leftarrow 0$ and $d\theta_v \leftarrow 0$.
 4:         Synchronize thread-specific parameters $\theta'_\pi = \theta_\pi$ and $\theta'_v = \theta_v$, set $t_{start} = t$, and get state $b_t$
 5:         repeat
 6:             Sample action $a_t$ according to policy $\pi(a_t \mid b_t;\theta_\pi)$
 7:             Take action $a_t$, receive reward $r_t$ and new belief $b_{t+1}$
 8:             $t \leftarrow t + 1$
 9:         until reach terminal $b_t$ or $t - t_{start} == t_{max}$
10:         For non-terminal $b_t$: $R \leftarrow V(b_t;\theta'_v)$; for terminal $b_t$: $R \leftarrow 0$
11:         for $i = t - 1$ to $t_{start}$ do
12:             $R \leftarrow r_i + \gamma R$
13:             Accumulate policy gradients w.r.t $\theta'_\pi$:
                $d\theta_\pi \leftarrow d\theta_\pi + \nabla_{\theta'_\pi} \log \pi(a_i \mid b_i;\theta'_\pi) A(a_i, b_i;\theta'_\pi, \theta'_v)$
                $+\beta\nabla_{\theta'_\pi} H(\pi(a_i \mid b_i;\theta'_\pi))$.
14:             Accumulate critic gradients w.r.t. $\theta'_v$:
                $d\theta_v \leftarrow d\theta_v + 2(R - V(b_i;\theta'_v))\nabla_{\theta'_v} V(b_i;\theta'_v)$
15:         end for
16:         Update $\theta$ and $\theta_v$ asynchronously according to (7)
17:     end for
18: end while
```

To efficiently solve the belief MDPs formulated from $\boldsymbol{P3}$ and $\boldsymbol{P5}$, we propose an A3C-based algorithm to train the actor and critic networks. In particular, we view the negative of per time step cost of $-\mathcal{C}_t(b_t, a_t)$ as a scalar reward of r_t received by the agent at time t. Consider that the A3C algorithm applies the mix of k-step returns to update both the policy and value-function [50], the reward function can be written as $R_t = \sum_{i=0}^{k-1} \gamma^i(-\mathcal{C}_{t+i}) + \gamma^k V(b_{t+k};\theta_v)$. Hence the advantage function can be approximated by $A(a_t, b_t;\theta_\pi, \theta_v) = R_t - V(b_t;\theta_v)$, where k changes from state to state and is no larger than t_{max}; the parameter $\gamma \in [0, 1]$ is the discount factor, which penalizes the rewards in the future so as to guarantee that the return over long episodes remains finite. According to the Policy Gradient Theorem [51], the update of parameter θ_π follows $\nabla_{\theta_\pi} \log \pi(a_t \mid b_t;\theta_\pi) A(a_t, b_t;\theta_\pi, \theta_v) + \beta\nabla_\theta H(\pi(b_t;\theta_\pi))$, where the extra entropy regularization term $H(\pi(b_t;\theta_\pi))$ is applied to encourage exploration, and the hyperparameter β is used to control the strength of the regularization. The parameter θ_v of the critic network is updated with the goal of iteratively minimizing the sequence of loss functions by using gradient descent, where the loss function at time t is defined as the mean-squared error (MSE) between R_t and $V(b_t;\theta_v)$, i.e., temporal difference, and written as $f_t(\theta_v) = (R_t - V(b_t;\theta_v))^2$. By differentiating the loss function $f_t(\theta_v)$ with respect to θ_v, we derive the update rule of the parameter θ_v as $\nabla_{\theta_v} f_v(\theta_v) = 2(R_t - V(b_t;\theta_v))\nabla_{\theta_v} V(b_t;\theta_v)$. In our implementation, the optimization algorithm for training the actor and critic networks is chosen as RMSProp [52], which is commonly used in deep learning. Specifically, the update functions in the standard RMSProp update are

$$g = \alpha g + (1 - \alpha)\Delta\theta^2, \theta \leftarrow \theta - \eta\frac{\triangle\theta}{\sqrt{g + \epsilon}}, \tag{7}$$

where α is the RMSProp decay factor, $\Delta\theta$ is the accumulated gradients, η is the learning rate, and ϵ is the exploration rate which is normally a small positive number.

In our algorithm, both actor and critic networks take belief state as inputs, and the actor outputs action probabilities while the critic outputs estimated value functions. The network's weights are updated by calculating the value loss for the critic and the policy loss for the actor and then backpropagate those errors. During the training process, the actor is learning to produce better action probabilities and the critic is becoming better at evaluating those actions. The details of generating the optimal PAMDRM based on A3C are summarized in Algorithm 1 [5]. We first initialize the actor $\pi(a_t|b_t;\theta_\pi)$ and the critic $V_\pi(b_t;\theta_v)$ in the global network. For each specific worker agent in parallel, we repeat: getting a copy of the global actor $\pi(a_t|b_t;\theta_\pi)$ and critic $V_\pi(b_t;\theta_v)$; sampling an episode of n steps; computing the accumulated gradients $d\theta_\pi$ and $d\theta_v$; updating the global actor and critic networks asynchronously. This process is repeated until convergence. When implementing Algorithm 1 to solve the belief MDPs formulated from $\boldsymbol{P3}$ and $\boldsymbol{P5}$, although their implementation procedures are the same, we should notice that their belief states, state update functions and cost functions are different, which should be implemented according to corresponding expressions in (2), (3), (4), (6).

[3] In most cases, the number of workers is the number of logical CPU cores in a computing device.

[4] For ease of notation, we use b_t and a_t to denote the belief state and action probabilities respectively.

[5] Note that the computation complexity of Algorithm 1 is dominated by the 7th step, where the reward r_t needs to be calculated in the environment for the MDPs according to (3) and (6). Specifically, the dominant part in calculating (3) and (6) is the calculation of the transition probability $p(o_t|o_{t-1})$. Theoretically, we have to enumerate over all the realizations of O_t and O_{t-1} to derive $p(o_t|o_{t-1})$, namely, the computation complexity is $O(|O_t|^2)$. In addition, the amount of working storage for Algorithm 1 is also dominated by the space for caching the value of $p(o_t|o_{t-1})$, hence the space complexity is $O(|O_t|^2)$. However, it is worth mentioning that for a real-world dataset, it is almost impossible that each realization of O_t will occur in practice due to the fact that users generally do not visit all the locations of a certain map. Therefore, we can apply our algorithm to a dataset where M and L could be potentially large since the computation of $p(o_t|o_{t-1})$ will only rely on those realizations with non-zero probabilities. More importantly, given a dataset, $p(o_t|o_{t-1})$ can be pre-trained offline and used as an input for Algorithm 1, thereby saving the computation time for its later usage.

VI. Experimental Evaluation

In this section, we evaluate and compare the performance of PAMDRM_user and PAMDRM_agg with two state-of-the-art privacy protection methods on synthetic and real-world datasets. We first introduce the experiment setup for implementing Algorithm 1 [6] and how we pre-process the dataset, followed by presenting and discussing the evaluation results.

A. Experiment Setup and Dataset Pre-processing

All experiments are implemented with Tensorflow in Python [53], and performed on a laptop with 2.6 GHz Intel Core i7-10750H CPU, 64GB RAM, 12 logical CPU cores. We create a specific environment for our RL agent by implementing the OpenAI gym interface [54]. Each experiment uses 12 actor-learner threads running on a single machine. All methods performs updates after every $t_{max} = 10$ actions and the maximum number of global shared counter T_{max} is set as 200 . The global maximum number of episodes is set as 2000. The learning rates for training the actor and critic networks are set as 0.0001 and 0.001 respectively [50]. All experiments used a discount factor of $\gamma = 0.9$, an RMSProp decay factor of $\alpha = 0.9$, and an exploration rate of $\epsilon = 1e - 10$. For the actor network, it takes belief states as input and the outputs action probabilities that sum to one. For the critic network, it also takes belief states as input and outputs a single value to evaluate the action selected by the actor network.

We use a real-world dataset Gowalla collected by Stanford University from a location-based social networking website [55], where users share their locations by checking-in. The check-in information contains user id, check-in time, latitude, longitude, and location id. Normally, we are more interested in the aggregate statistics in a certain region. For simplicity, we take the area of San Francisco as an example to describe how we pre-process the dataset. Specifically, we first filter the dataset to contain the location check-in in San Francisco, and convert the latitude and longitude coordinates into grid coordinates by a unit of 0.01 (44 meters approximately), remaining 238 unique locations in total. After that, we rank the users in decreasing order of their total number of checking-ins and select the top 50 users' data, obtaining a dataset that contains 36122 location check-ins in total and 150 unique locations. For simplicity, we set the time granularity to one hour, and consider a one-week aggregate mobility data from September 1st to 7th, 2010. Therefore, we can sample a Markov chain of O_t with the length of $7 \cdot 24 = 168$, with 72 unique realizations of O_t, namely $|O_t| = 72$. Next, we train its Markov transition matrix by calculating $p(o_t|o_{t-1}) = \frac{Count(o_t, o_{t-1})}{Count(o_{t-1})}$, where t takes value from $\{1, 2, ..., 168\}$, $Count(\cdot, \cdot)$ and $Count(\cdot)$ represent the number of occurrences of two realizations and a single realization respectively. Similarly, we also train the Markov transition probabilities $p(u_t|u_{t-1}) = \frac{Count(u_t, u_{t-1})}{Count(u_{t-1})}$ for the 50 users that we obtained.

B. Evaluation of Convergence

We first show the performance of convergence of Algorithm 1 on generating PAMDRM_agg and PAMDRM_user in Fig. 4, where we set $|O_t| = 20$ for simplicity and the distortion at time step t is set as $D_t = 5$, i.e., the total distortion is $D = 5 * T = 1000$. The 20 realizations are randomly selected from the 72 unique realizations that are sampled from the 50 users' data in the San Francisco area, which has been discussed in the previous sub-section. The results in Fig. 4 show that both PAMDRM_user and PAMDRM_agg converge within 2000 episodes. Specifically, their converged values are around 72 and 83 respectively, which validates the theoretical result in Theorem 3 that $\mathcal{L}^*_{user}(D)$ is upper bounded by $\mathcal{L}^*_{agg}(D)$.

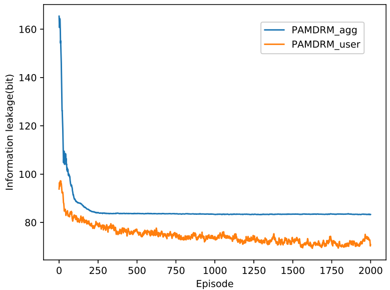

Fig. 4. Convergence of the proposed release mechanisms.

C. Evaluation of Privacy Protection

As discussed in the Introduction Section, we should evaluate the privacy protection level provided by privacy-preserving mechanisms from different dimensions to avoid possible over-estimation of privacy guarantee. For that reason, we perform the privacy evaluation of PAMDRM_user, PAMDRM_agg, and two other state-of-the-art privacy protection mechanisms proposed in [1] and [2] in terms of mutual information leakage and adversarial success on synthetic and real-world datasets.

1) Comparison Methodology: We set up the first compared mechanism by extending the context-aware localized mechanism PDPR proposed in [1] to our aggregate mobility data release setting [7]. This extension is implemented by following the design methodology discussed in Section VII-B2 and denoted by EoPDPR. The second compared one is applying a differentially private centralized mechanism to release aggregate mobility data. In particular, we select the most relevant one that is studied in [2][8], called discrete

[6] Via Tensorflow's built-in module TensorBoard, we show the visualization of the architecture of Algorithm 1 in Fig. 1 in Appendix, which contains a global network and 12 worker agents (threads), and starts with constructing the global network, followed by propagating the parameters of the global network synchronously to each worker agent. Fig. 2 in Appendix shows W_8's internal implementation structure with its own network and environment, and updates the global network parameters by interacting with its environment.

[7] We want to emphasize that the following comparison results are derived in a multi-user scenario and PDPR has been proved to be the optimal privacy-preserving release mechanism in protection single user's location trace.

[8] We did not compare our method with the mechanisms studied in [16] due to the difficultly in deriving the conditional probability distribution $p(r_t|o_t)$. It is non-trivial since the noise added on each element of a vector is not an integer, so it is almost impossible to count the co-occurrence of the realizations of (r_t, o_t) to calculate $p(r_t|o_t)$. In addition, to calculate mutual information in (1), we need to enumerate over all the realizations of both o_t and r_t, which is infeasible because they are continuously valued. It is noteworthy that the overall differential privacy guarantee is $O(|L| \cdot |T| \cdot \epsilon)$ for an aggregate matrix that is released by the mechanism studied in [16] since it achieves ϵ-DP for each element in the matrix. However, according to the composition theorem [19], DMLM is overall $O(|T| \cdot \epsilon)$ differentially private for releasing an aggregate matrix since it achieves ϵ-DP for each vector in the matrix. The work in [56] is also incomparable for a similar reason as discussed for [16]. Liu et al. [57] designed a DP mechanism that considers user's ID, location, timestamps at the same time for a different problem setting from ours, which could be an interesting research direction in our future work.

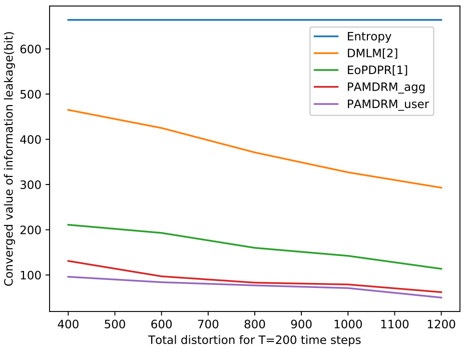

Fig. 5. Comparison of privacy-utility tradeoffs of PAMDRM_user, PAMDRM_agg, EoPDPR, and DMLM when $T = 200, |O_t| = 20$.

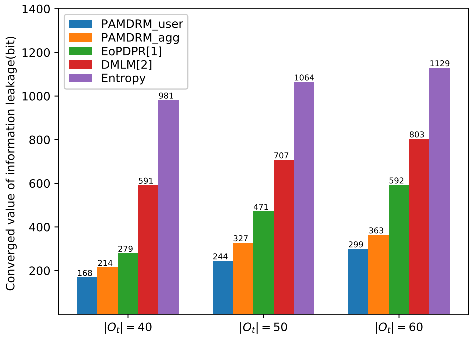

Fig. 6. Comparison of converged value of information leakage of PAMDRM_user, PAMDRM_agg, EoPDPR, DMLM, and entropy of O_T under different number of $|O_t|$ when $T = 200, D_t = 5$.

multidimensional Laplacian mechanism (shorted as DMLM), which adds independent Laplacian noise to each component of the input vector (i.e., a realization of o_t in our problem), and satisfies $(\epsilon, 0)$-differential privacy. The probability mass function $\mathcal{P}$ of DMLM is defined as

$$\mathcal{P}(\boldsymbol{i}) = \left(\frac{1-\lambda}{1+\lambda}\right)^d \lambda^{|i_1|+|i_2|+\ldots+|i_d|}, \quad \forall \boldsymbol{i} \in \mathbb{Z}^d, \tag{8}$$

where $\boldsymbol{i}$ is the noise vector, $\lambda \triangleq e^{-\frac{\epsilon}{\Delta}}$ and ϵ is the privacy budget, Δ is the sensitivity of a query function (equals to 1 in our problem), and d is the dimension of vector $\boldsymbol{i} = (i_1, i_2, ..., i_d)$. In addition, as proved in [2], the L2 norm distortion function for DMLM can be estimated by $D = \frac{2d\lambda}{(1-\lambda)^2}$.

2) Evaluation of Information Leakage: We consider an aggregation process of $T = 200$ time steps. For all compared methods, we use the same distortion D_t, location priors, and the previous trained Markov transition matrix as the inputs in our algorithm. To measure the user-level information leakage $I(U_m^T; R^T)$ of user-m (i.e., the objective in $\boldsymbol{P2}$) for EoPDPR, we need to know the conditional probability distribution $p(r|u_m)$, which can be estimated as the following. Firstly, each user samples an instance of perturbed trace according to her local mechanism $p(v_m|u_m)$ (shown in Fig. 13), and then all their perturbed traces are aggregated to obtain an instance of the perturbed aggregate location R^T. We run this process for 1000 times, count the co-occurrence of a certain pair of (r, u_m), and the normalized frequency is used as an estimation of $p(r|u_m)$. To implement DMLM and compute its user-level information leakage, we first set d as the dimension of O_t. At each time step t and for a fixed D_t , we enumerate ϵ from 0.0001 to 10 with an increase of 0.0001, and use the first ϵ that satisfies the distortion constraint as the privacy budget. Once we obtain an ϵ, we can derive the mechanism $p(r_t|o_t)$ by setting $\boldsymbol{i} = r_t - o_t$ in (8), and similar as before we also run this mechanism for 1000 times, and calculate the normalized frequency to approximate $p(r|u_m)$. For the following time steps, we compute the current probability distribution $p(o_t)$ based on equation $p(o_t) = \sum_{o_{t-1}} p(o_t|o_{t-1})p(o_{t-1})$, and then repeat the process for time t. Note that we add a noise vector $\boldsymbol{i}$ to O_t independently at each time step, the total leakage of DMLM equals to the summation of the above per-time-step user-level information leakage over $T = 200$ time steps.

The evaluation results on different number of users are shown in Fig. 5. We can see that Individual Privacy – Aggregate Utility tradeoff is upper bounded by the Aggregate Privacy – Aggregate Utility tradeoff, while along with the increase in distortion, their leakages tend to be close to each other. Clearly, this observation coincides with the theoretical result as proved in Theorem 3. It is worth noticing that both PAMDRM_user and PAMDRM_agg perform better than EoPDPR and DMLM in terms of mutual information leakage, with DMLM incurring the largest leakage. The reason is that using mutual information as the privacy metric for aggregate mobility data naturally considers the prior distributions and temporal correlations among users' traces in mechanism design to ensure high privacy-utility tradeoff by adding selective noise according to the prior information and correlation. However, as a context-free metric, DP does not have such a property. Besides, theoretically speaking, all the information leakage should be no larger than the entropy of O^T since $I(O^T; R^T) = H(O^T) - H(O^T|R^T) \le H(O^T)$ holds, which is also validated by the results presented in Fig. 5, where the symbol 'entropy' represents $H(O^T)$.

Lastly, to investigate how the number of realizations $|O_t|$ affect the privacy leakage, we present evaluation results corresponding to $|O_t| = 40, 50, 60$ under the distortion constraint $D_t = 5$. The results are shown in Fig. 6, where 'user' and 'agg' in the x-axis denote the user-level and aggregate-level leakage respectively. We can also see that PAMDRM_user incurs the least leakage, followed by PAMDRM_agg and EoPDPR, and DMLM incurs the highest leakage, all of which are upper bound by $H(O^T)$. Additionally, the leakage becomes larger with the increase in the number of realizations. The reason can possibly be that larger datasets tend to be more informative, thereby leaking more information about users and the datasets themselves. Even so, we are aware that, intuitively, the user-level leakage may decrease with the increase in $|O_t|$, since each user's data is hidden in a larger realization set of O_t, which makes it harder to deduce her own data. However, we want to highlight that the results in Fig. 6 are derived subject to a fixed distortion of $D_t = 5$ (total distortion is $D = 1000$), so the release mechanism will leak more information to satisfy the distortion constraint. To verify this intuition, we impose a larger distortion as $D_t = 10$ ($D = 2000$ in total) when $|O_t| = 50$, we get the user-level leakage as 154, which is

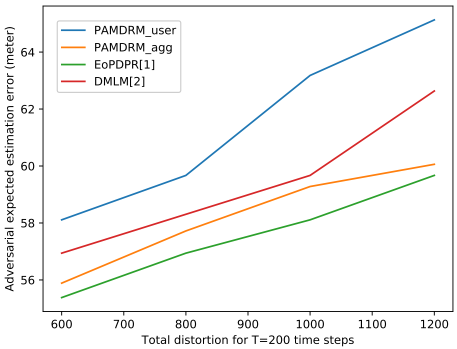

Fig. 7. Adversary's success under expected estimation error on real-world dataset.

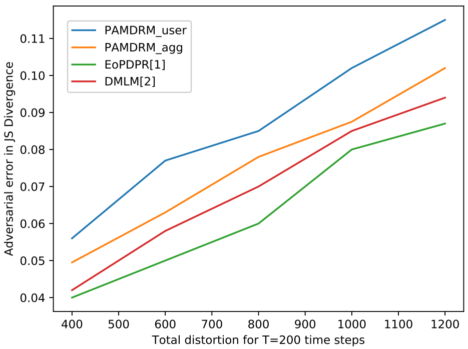

Fig. 8. Adversary's success under Jensen-Shannon Divergence on real-world dataset.

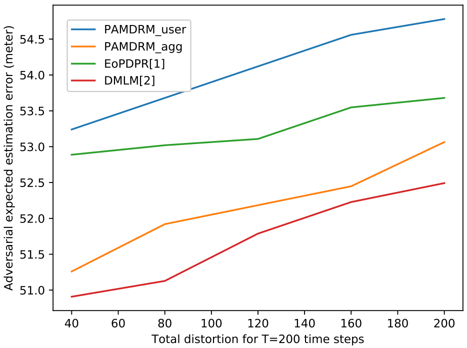

Fig. 9. Adversary's success under expected estimation error on synthetic dataset.

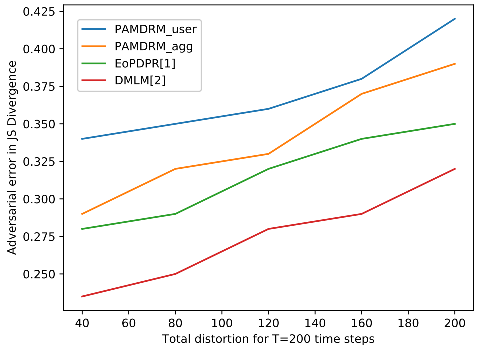

Fig. 10. Adversary's success under Jensen-Shannon Divergence on synthetic dataset.

smaller than 171 when $|O_t| = 40, D_t = 20$.

3) Evaluation of Attack Success: To have a better understanding of the privacy protection performance provided by these mechanisms in addition to information leakage, we also evaluate the adversarial success. Specifically, we perform the well-principled Bayesian updating location inference attack [16] on the released noisy aggregate location, and then evaluate the adversary's success on the worst-case user by calculating her error defined by the Jensen-Shannon (JS) Divergence [16] and the expected estimation error [34].

Next, we describe how to perform the inference attack on these mechanisms. Essentially, performing a Bayesian updating attack relies on computing the user's posterior probability distribution $\hat{P}_u$ for each time step t, given the prior knowledge on the user's location distribution P_u and the released aggregate location $\boldsymbol{R}$. After the adversary obtains the posteriors from the Bayesian updating attack under different mechanisms, we calculate the expected estimation error averaging over all time steps and the JS Divergence-based error for each mechanism separately. Specifically, adversary's expected estimation error is defined as $\sum_x \hat{P}_u \cdot ||x-u||$, where u and x are the target user's true location and adversary's inferred location with non-zero probability in her posterior knowledge $\hat{P}_u$, respectively, and represented by two-dimensional coordinates; $||x-u||$ is the Euclidean distance between x and u. The JS Divergence-based error is defined as $\frac{\sum_{t\in T} JS(P_u||\hat{P}_u)}{T}$, where JS takes values in the range of $[0,1]$, and represents the distance of distributions between adversary's inferred $\hat{P}_u$ and user's true location prior P_u. As discussed in [16], this error measures the adversary's success in terms of her mean error on the inference attack over time T. In particular, the adversary is assumed to know users' location prior distribution P and we select the worst-case user as the target of the Bayesian updating inference attack. Given a certain distortion D, EoPDPR, DMLM, PAMDRM_agg, and PAMDRM_user output their own perturbed aggregate location $\boldsymbol{R}$ separately, which are then combined with P by the adversary to perform Bayesian inference attack and eventually get the corresponding four posterior probability distributions $\hat{P}_u$ of that target user. After that, we can calculate adversarial success regarding the expected estimation error and the JS-based error based in these posteriors $\hat{P}_u$ and her prior knowledge P_u. In particular, the higher the adversarial success is, the better privacy protection a mechanism can provide.

We first analyze the results of $M = 20$ users' traces and their top $L = 10$ locations from the real-world dataset, as shown in Fig. 7 and 8. As we can see, the error of adversary's inference increases with larger allowable distortion in all mechanisms. The rationale is that larger distortion allows the mechanisms to add more noise to strengthen privacy protection, resulting in higher estimation error. In addition, PAMDRM_user has the highest adversary error while EoPDPR has the lowest error, with PAMDRM_agg and DMLM in between. We want to highlight that their major difference lies in mechanism design, i.e., PAMDRM_agg, PAMDRM_user, and DMLM are designed in a centralized manner as illustrated in Fig. 1, while EoPDPR is a direct extension from a local mechanism as shown in Fig. 13. Typically, a localized mechanism introduces more noise than required in the centralized setting under the same privacy level [58], [59]. In other words, the centralized mechanism can provide better privacy protection under the same distortion constraint by using all users' location trace as input and adding noise directly on the aggregated data, which supports the above numerical results.

Additionally, we also evaluate adversary's success on a synthetic dataset with $M = 2$ users moving within $L = 2$ locations in Fig. 9 and 10. Specifically, their location prior distributions are assigned to $\{0.9, 0.1\}$ and $\{0.8, 0.2\}$, and both their transition probabilities are $\{0.9, 0.1; 0.9, 0.1\}$. Results show the same trend in privacy-utility tradeoffs as in Fig. 5, so we omit it due to space consideration. Interestingly, we notice that EoPDPR has a larger adversary error than DMLM on both measures, in contrast to previous results shown in Fig. 7 and 8. This indicates that the number of users may affect the performance of centralized mechanisms against adversary's inference attack. Note that the priors and transitions on the synthetic dataset are highly skewed, making the advantage of using a context-aware metric in EoPDPR dominant in privacy protection even though it is a localized mechanism. In other words, when the number of participants in an aggregation process is small, the advantage of using context-aware metrics for aggregate mobility data outweighs the disadvantage of using a localized mechanism, due to the fact that an individual's prior information has more effect on a dataset of a smaller number of users than a larger one. Therefore, it is reasonable that EoPDPR leads to a lower adversarial success to perform inference attacks compared with DMLM on a small number of users, but DMLM still has its advantage in lower adversarial success when a large number of users participate in the aggregation.

Finally, we can see that the overall performance of PAMDRM_user remains the best since it utilizes the context-aware metric and is designed in a centralized manner. To be more specific, PAMDRM_user considers location priors and temporal correlations and therefore provides more protection than DMLM under the same distortion by adding selective noise according to the data priors to guarantee high utility-privacy tradeoffs. More importantly, it also has low adversary's success, which is reasonable because even though mutual information protects average inference attack by definition, the objective function also captures the worst-case individual user-level leakage, matching the goal of the adversary's inference attack. Another important insight is that even though PAMDRM_agg achieves better privacy-utility tradeoff than the other two, it does not always perform better than them on attack success and it may be because the goal of Bayesian attack is on individual user $\boldsymbol{U}_m$ while PAMDRM_agg aims to protect the original location aggregate $\boldsymbol{O}$. This also highlights the necessity of considering privacy measures from different perspectives to avoid overestimating the level of privacy offered to the user, concurred with the insight given in [3].

VII. Discussions

In this section, we first provide a thorough discussion on the design of a data-driven based scheme for our privacy-preserving aggregate mobility data release problem by leveraging a recently proposed effective mutual information estimator. Next, we introduce alternative formulations for this release problem including average and joint user-level leakage, together with two interesting extensions of [1] in detail.

A. A Data-driven Approach

Experimental results have validated the our proposed privacy-preserving aggregate mobility data release mechanism can achieve a better privacy-utility tradeoff by using a context-aware metric, which can be efficiently obtained by using an RL approach. However, it is not applicable to the scenario where close-form location priors and temporal correlation do not exist since we only have limited data samples due to inaccurate or missing data. To tackle this issue, we discuss a data-driven approach to derive the optimal PAMDRM in terms of minimizing mutual information privacy leakage under a distortion constraint. Especially, we provide an adversarial training framework following the idea in [60], [61] to design a data-driven based scheme by learning from users' location trace samples without requiring explicit location prior and transition probabilities

Recently, using the adversarial training framework to design privacy-preserving data release mechanisms has been studied in [60], [61]. However, the adversarial training framework formulation and experimental evaluation in [60] can only be applied to binary data due to the difficulty in calculating the gradient of loss functions. Tripathy et al. [61] overcome this drawback by estimating a lower bound of mutual information to connect their proposed framework to a general mutual information estimator, which is applicable to arbitrary type of data structure. Nevertheless, the focus of previous work was mainly on mutual information lower bound approximation [27], [61], [62]. Considering their inconsistency in mutual information minimization problems since a lower bound cannot be used for minimization, we adopt a novel upper bound of mutual information proposed by Cheng et al. [40] as the approximation of mutual information, which has been validated to be reliable and more effective when applied to mutual information minimization tasks therein. Note that Cundy et al. [63] also use an upper bound as an estimation of mutual information to guarantee that the policy derived from an RL algorithm satisfies the privacy constraint, but their model-free mutual information estimator is a loose upper bound. We want to highlight that [40] does not use an adversarial training framework because their goal is to minimize the mutual information between two given datasets, while we aim at finding the optimal privacy-preserving mechanism to generate noisy location aggregates and the training data samples are only users' location traces.

Motivated by the work studied in [60], [61], we transform the Individual Privacy – Aggregate Utility tradeoff in $\boldsymbol{P2}$ into the following unconstrained minimax optimization problem,

$$\min_{Q(r|o)} \max_{A(u|r)} \mathbb{E}[\log A(U|R)] + \lambda \mathbb{E}[d(O, R)], \tag{9}$$

where U, O, R denote user's location trace, original location aggregate, and perturbed aggregate; $Q(R|O)$ and $A(U|R)$

represent the privacy-preserving mobility data release mechanism and the adversary's likelihood; the parameter λ is the Lagrange multiplier, and we can use smaller λ to minimize information leakage over distortion, and vice versa. Following the design methodology for adversarial network studied in [61], we provide an adversarial training framework to solve the above minimax optimization problem, as shown in Fig. 11 [9]. By leveraging this framework, we can train a PAMDRM to minimize both the distortion and privacy loss terms, and train the adversary to maximize the privacy loss. Our privacy mechanism PAMDRM $Q(R|O)$ and the adversary's likelihood $A(U|R)$ are represented by neural networks parameterized using θ and ϕ, respectively, and taking users' traces $U_1, ..., U_M$ as input data samples. Specifically, the privacy loss is defined as $\mathbb{E}[\log A(U|R)] = H(U|R) = H(U) - I(U;R)$. From the adversary's perspective, maximizing the privacy loss is equivalent to minimizing $I(U;R)$ since $H(U)$ is a constant. As the key to our data-driven framework, we choose to estimate $I(U;R)$ via the upper bound given by [40], where the estimator has been proved to be more reliable and effective than previous works. Another advantage of this mutual information estimator is that it is trainable in gradient-descent frameworks, which guarantees effective and efficient training of the networks in a gradient-descent manner. Implementation of this scheme is considered as future work.

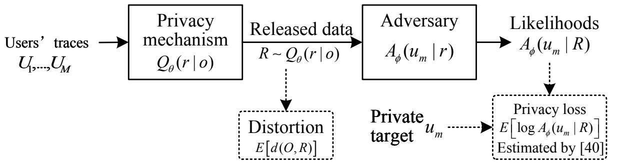

Fig. 11. Adversarial training framework for privacy-preserving aggregate mobility data release.

B. Alternative Formulations: Average and Joint User-level Leakage

In this subsection, we discuss another two potential mechanism designs for privacy-preserving aggregate mobility data release regarding average user leakage and a joint consideration of all users' traces, respectively. We use this discussion to strengthen our motivation for choosing the worst-case user-level leakage as the privacy metric in $\boldsymbol{P}2$. Additionally, two interesting extensions of [1] are presented, facilitating the discussion of our comparison methodology in Section VI.

1) Discussion about Applicability: An intuitive definition for user-level leakage is the average value, as shown in Definition 5. Note that the optimal mechanism in terms of worst-case user-level leakage in $\boldsymbol{P}2$ can guarantee that each user's privacy leakage is upper bounded, i.e., no larger than the maximal leakage that occurred among all users. However, the optimal mechanism in terms of average user-level leakage in Definition 5 cannot provide such a guarantee, since minimizing average leakage does not mean each user's leakage is minimized, and therefore certain users' leakage can be much larger. Hence, average leakage is not suitable for designing mechanisms that target on protecting individual user's leakage, while our goal is to provide privacy guarantee for each user.

***Definition** 5:* **Average User-level Leakage:** *In an online aggregate mobility data release setting, given utility constraint D_t at time step t, the privacy-utility tradeoff is defined as*

$$\mathcal{L}^*_{average}(D) = \min_{\substack{q_t(r_t|o^t, r^{t-1}):\\ \{D(O_t,R_t)\le D_t\}_{t=1}^T}} \frac{\sum_{1\le m\le M} I(U_m^T; R^T)}{M}.$$

Another type of centralized mechanism design takes all users' traces into account by treating the concatenation of multiple users' traces as joint multi-variate random variables and outputs perturbed aggregated result based on all of them, and the privacy-preserving release mechanism is in the form of $q(r^T|u_1^T, ..., u_M^T)$. This model is illustrated in Fig. 12, and the notation U^T and R^T are simplified as U and R. We formulate the above release problem in Definition 6. Intuitively, this formulation makes more sense when all users' traces are correlated, since the correlation can be captured naturally in the joint distribution of $p(u_1^T, ..., u_M^T)$. Even so, our problem setting assume users to be independent, so it is reasonable to not adopt this type of formulation due to the complexity issue caused by the high dimension of vector $U_1^T, U_2^T, ..., U_M^T$, but we consider it as interesting future work.

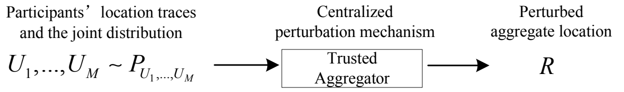
Participants' location traces and the joint distribution
$U_1,...,U_M \sim P_{U_1,...,U_M}$
Centralized perturbation mechanism
Trusted Aggregator
Perturbed aggregate location
R

Fig. 12. Privacy-preserving aggregate mobility data release with concatenation of users' traces.

***Definition** 6:* **Joint User-level Leakage:** *In an online aggregate mobility data release setting, given the utility constraint D_t at time step t, the tradeoff is defined as*

$$\mathcal{L}^*_{joint}(D) = \min_{\substack{q(r^T|u_1^T,...,u_M^T):\\ \{D(O_t,R_t)\le D_t\}_{t=1}^T}} I(U_1^T, U_2^T, ..., U_M^T; R^T),$$

where $u_1^T, ..., u_M^T$ represent each user's trace, O_t and R_t are the original and perturbed aggregate, and $q(r^T|u_1^T, ..., u_M^T)$ is the privacy-preserving mobility data release mechanism.

2) Extension of [1] to Aggregate Mobility Data: Even though the privacy-aware location trace release mechanism called PDPR [1] was originally designed for privacy-preserving location trace release for an individual user, it would be interesting to extend it to our problem setting. A straightforward extension illustrated in Fig. 13 is to let each participant perform PDPR locally with her original trace U_m to obtain noisy trace V_m, and then upload V_m to an aggregator, who can be either trusted or untrusted. After receiving all the M users' perturbed traces, the aggregator adds them up and release the perturbed aggregate location $R = V_1 + ... + V_M$.

Another way to extend is to treat the concatenation of multiple users' traces as joint multi-variate random variables and output perturbed aggregated result based on all of them, which essentially adopts the idea of mechanism design in Fig. 6. Note that this is a centralized approach, but it introduces exponential

[9] For space consideration, the discussion of the data-driven approach to solve $\boldsymbol{P1}$ is omitted since we can replace the users' traces samples with the original aggregate location O^T as input.

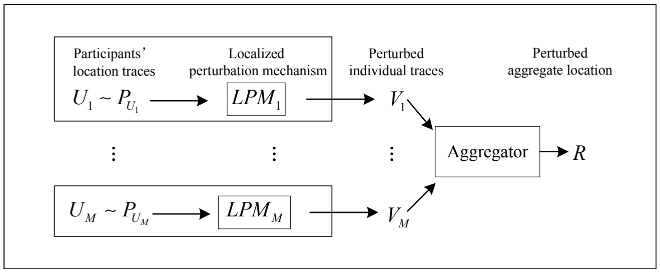
Participants' location traces
Localized perturbation mechanism
Perturbed individual traces
Perturbed aggregate location
$U_1 \sim P_{U_1}$
LPM_1
V_1
Aggregator
R
$U_M \sim P_{U_M}$
LPM_M
V_M

Fig. 13. Privacy-preserving aggregate mobility data release via local perturbation mechanism.

computational complexity since all the users' locations form a high dimensional vector, so we only performed the evaluation of the local extension of [1] in Section VI rather than this centralized one. Thus, we choose to perturb the aggregated locations in this paper, which is why our release mechanism is centralized and in the form of $q(r^T|o^T)$ as shown in Fig.1. More importantly, a centralized mechanism can also provide better privacy protection under the same distortion constraint by using all users' location trace as input and adding noise directly on the aggregated data [58], [59], which has been shown in the numerical results in Section VI.

VIII. Concluding Remarks

In this paper, the problem of privacy-preserving aggregate mobility data release was investigated by using an information-theoretic approach, and a practical algorithm was proposed by adapting the efficient A3C approach based on the MDP formulations of our proposed privacy problems. Experimental results show that optimal release mechanism regarding minimizing worst-case user-level leakage performs better compared with two state-of-the-art privacy protection methods in terms of both privacy-utility tradeoff and adversarial success since it adopts a context-aware metric and is a centralized design. We also found the number of users affects the privacy protection performance of a centralized mechanism. More importantly, results also indicated that we should consider privacy measures from different dimensions to avoid misjudging the privacy guarantee provided to users. Finally, as an important and promising future work, we will further study and implement the adversarial training framework designed for our privacy-preserving aggregate mobility data release problem and validate its effectiveness as a fully data-driven approach.

References

[1] E. Erdemir, P. L. Dragotti, and D. Gündüz, "Privacy-aware time-series data sharing with deep reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 389–401, 2020.

[2] Q. Geng and P. Viswanath, "Optimal noise adding mechanisms for approximate differential privacy," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 952–969, 2015.

[3] S. Oya, C. Troncoso, and F. Pérez-González, "Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1959–1972.

[4] C. O. Buckee, S. Balsari, J. Chan, M. Crosas, F. Dominici, U. Gasser, Y. H. Grad, B. Grenfell, M. E. Halloran, M. U. Kraemer *et al.*, "Aggregated mobility data could help fight covid-19," *Science (New York, NY)*, vol. 368, no. 6487, pp. 145–146, 2020.

[5] E. Pollina and D. Busvine, "European mobile operators share data for coronavirus fight," *Reuters https://www. reuters. com/article/us-health-coronavirus-europe-telecoms/european-mobile-operators-share-data-for-coronavirus-fight-idUSKBN2152C2*, 2020.

[6] (2020) Apple makes mobility data available to aid covid-19 efforts. [Online]. Available: https://www.apple.com/newsroom/2020/04/apple-makes-mobility-data-available-to-aid-covid-19-efforts/

[7] J. Lopes, J. Bento, E. Huang, C. Antoniou, and M. Ben-Akiva, "Traffic and mobility data collection for real-time applications," in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 216–223.

[8] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," in *International Conference on Pervasive Computing*. Springer, 2009, pp. 390–397.

[9] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Measuring membership privacy on aggregate location time-series," in *Abstracts of the 2020 SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems*, 2020, pp. 73–74.

[10] P. Maas, S. Iyer, A. Gros, W. Park, L. McGorman, C. Nayak, and P. A. Dow, "Facebook disaster maps: Aggregate insights for crisis response & recovery." in *KDD*, vol. 19, 2019, p. 3173.

[11] Waze, https://www.waze.com.

[12] Uber, https://movement.uber.com/.

[13] E. Creaco, P. Kossieris, L. Vamvakeridou-Lyroudia, C. Makropoulos, Z. Kapelan, and D. Savic, "Parameterizing residential water demand pulse models through smart meter readings," *Environmental Modelling & Software*, vol. 80, pp. 33–40, 2016.

[14] C. F. P. Bureau, "Consumer protection principles: Consumer-authorized financial data sharing and aggregation," *October*, vol. 18, p. 2017, 2017.

[15] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, "Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1241–1250.

[16] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "What does the crowd say about you? evaluating aggregation-based location privacy," in *Proceedings on Privacy Enhancing Technologies*. Springer, 2017, pp. 76–96.

[17] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Knock knock, who's there? membership inference on aggregate location data," *arXiv preprint arXiv:1708.06145*, 2017.

[18] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[19] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM, 2010, pp. 715–724.

[20] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 3, p. 26, 2011.

[21] S.-S. Ho and S. Ruan, "Differential privacy for location pattern mining," in *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*. ACM, 2011, pp. 17–24.

[22] G. Acs and C. Castelluccia, "A case study: Privacy preserving release of spatio-temporal density in paris," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 1679–1688.

[23] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," in *Entropy*, 2017.

[24] B. Jiang, M. Li, and R. Tandon, "Context-aware data aggregation with localized information privacy," in *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018, pp. 1–9.

[25] W. Zhang, M. Li, R. Tandon, and H. Li, "Online location trace privacy: An information theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 235–250, 2018.

[26] Q. Li, G. Cao, and T. F. La Porta, "Efficient and privacy-aware data aggregation in mobile sensing," *IEEE Transactions on dependable and secure computing*, vol. 11, no. 2, pp. 115–129, 2014.

[27] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 2180–2188.

[28] B. Jiang, M. Seif, R. Tandon, and M. Li, "Context-aware local information privacy," *IEEE Transactions on Information Forensics and Security*, 2021.

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

[30] F. Tramèr, Z. Huang, J.-P. Hubaux, and E. Ayday, "Differential privacy with bounded priors: reconciling utility and privacy in genome-wide association studies," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1286–1297.

[31] M. Bloch, O. Günlü, A. Yener, F. Oggier, H. V. Poor, L. Sankar, and R. F. Schaefer, "An overview of information-theoretic security and privacy: Metrics, limits and applications," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 5–22, 2021.

[32] G. Theodorakopoulos, R. Shokri, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Prolonging the hide-and-seek game: Optimal trajectory privacy for location-based services," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, 2014, pp. 73–82.

[33] R. Shokri, G. Theodorakopoulos, and C. Troncoso, "Privacy games along location traces: A game-theoretic framework for optimizing location privacy," *ACM Transactions on Privacy and Security (TOPS)*, vol. 19, no. 4, pp. 1–31, 2016.

[34] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *2011 IEEE symposium on security and privacy*. IEEE, 2011, pp. 247–262.

[35] F. du Pin Calmon and N. Fawaz, "Privacy against statistical inference," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2012, pp. 1401–1408.

[36] L. Sankar, S. R. Rajagopalan, and H. V. Poor, "Utility-privacy tradeoffs in databases: An information-theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 838–852, 2013.

[37] I. Issa, A. B. Wagner, and S. Kamath, "An operational approach to information leakage," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1625–1657, 2019.

[38] S. Saeidian, G. Cervia, T. J. Oechtering, and M. Skoglund, "Optimal maximal leakage-distortion tradeoff," *arXiv preprint arXiv:2105.01033*, 2021.

[39] S. Li, A. Khisti, and A. Mahajan, "Information-theoretic privacy for smart metering systems with a rechargeable battery," *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3679–3695, 2018.

[40] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, "Club: A contrastive log-ratio upper bound of mutual information," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1779–1788.

[41] L. Fan and L. Xiong, "Real-time aggregate monitoring with differential privacy," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 2169–2173.

[42] C. Kopp, M. Mock, and M. May, "Privacy-preserving distributed monitoring of visit quantities," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, 2012, pp. 438–441.

[43] R. A. Popa, A. J. Blumberg, H. Balakrishnan, and F. H. Li, "Privacy and accountability for location-based aggregate statistics," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 653–666.

[44] A. Pyrgelis, E. De Cristofaro, and G. J. Ross, "Privacy-friendly mobility analytics using aggregate location data," in *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2016, pp. 1–10.

[45] M. Götz, S. Nath, and J. Gehrke, "Maskit: Privately releasing user context streams for personalized mobile applications," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 2012, pp. 289–300.

[46] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1298–1309.

[47] K. Murphy. (1998) A brief introduction to reinforcement learning. [Online]. Available: https://www.cs.ubc.ca/~murphyk/Bayes/pomdp.html#POMDP

[48] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[49] D. Silver, "University college london course on reinforcement learning," https://www.davidsilver.uk/teaching/, 2015.

[50] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[51] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[52] T. Tieleman and G. Hinton, "Lecture 6.5- rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, 2012.

[53] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[54] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *CoRR*, vol. abs/1606.01540, 2016. [Online]. Available: http://arxiv.org/abs/1606.01540

[55] E. Cho, S. Myers, and J. Leskovec, "Friendship and mobility: Friendship and mobility: User movement in location-based social networks," *Proc. ACM SIGKDD 2011*.

[56] L. Ou, Z. Qin, S. Liao, Y. Hong, and X. Jia, "Releasing correlated trajectories: Towards high utility and optimal differential privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 1109–1123, 2018.

[57] B. Liu, S. Xie, H. Wang, Y. Hong, X. Ban, and M. Mohammady, "Vtdp: Privately sanitizing fine-grained vehicle trajectory data with boosted utility," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[58] T. H. Chan, E. Shi, and D. Song, "Optimal lower bound for differentially private multi-party aggregation," in *European Symposium on Algorithms*. Springer, 2012, pp. 277–288.

[59] B. Jiang, M. Li, and R. Tandon, "Local information privacy and its application to privacy-preserving data aggregation," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[60] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," *Entropy*, vol. 19, no. 12, p. 656, 2017.

[61] A. Tripathy, Y. Wang, and P. Ishwar, "Privacy-preserving adversarial networks," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2019, pp. 495–505.

[62] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *International Conference on Machine Learning*. PMLR, 2018, pp. 531–540.

[63] C. Cundy and S. Ermon, "Privacy-constrained policies via mutual information regularized policy gradients," *arXiv preprint arXiv:2012.15019*, 2020.