Pricing Utility vs. Location Privacy: A Differentially Private Data Sharing Framework for Ride-on-Demand Services

Zhirun Zheng, Zhetao Li, Saiqin Long, Suiming Guo, Chao Chen, *Senior Member, IEEE*, and Ke Xu, *Fellow, IEEE*

***Abstract*—Noise perturbation introduced by differential privacy (DP) could degrade the quality of essential services like dynamic pricing and ride-matching in ride-on-demand (RoD) services. In this paper, we focus on RoD services under an honest-but-curious server, and propose a Pricing-Aware Differentially Private framework (PADP-RoD) to protect users' location privacy while providing them with high-quality location-based services. Specifically, given that a price multiplier is subject to abrupt changes in response to shifts in supply and demand, especially near hotspots, we propose an adaptive supply and demand aware grid to capture the changes. Powered by the grid, we put forward two utility metrics for quantifying the quality loss of dynamic pricing and ride-matching services caused by perturbation, respectively. With those metrics, PADP-RoD is formulated as a minimization problem, aiming to minimize the quality loss of services given DP constraint. In this way, we can achieve an optimal balance between privacy and service quality. Due to the problem being a multi-objective optimization, we decompose it into a dynamic-pricing utility sub-problem and a ride-matching utility sub-problem, and solve them separately. To solve the dynamic pricing utility sub-problem, we propose a heuristic algorithm named the dynamic pricing mapping algorithm. Since the semi-infinite and non-differentiable nature of the ride-matching utility sub-problem, we transform this sub-problem into an unconstrained problem by the exact penalty function method, and solve it employing the particle swarm optimization algorithm. Our theoretical analysis demonstrates that PADP-RoD satisfies both ε_d-DP and ε_d-identifiability, and extensive experiments on a real-world dataset show that it can provide high-quality dynamic pricing and ride-matching services.**

***Index Terms*—Location privacy, ride-on-demand (RoD) services, dynamic pricing utility, differential privacy, ride-matching utility.**

Received 25 September 2023; revised 16 October 2024; accepted 17 January 2025. Date of publication 22 January 2025; date of current version 11 July 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62172350, Grant 62032020, Grant 62002135, Grant 62076214, Grant 62002135, and Grant U23B2027, in part by the National Key Research and Development Program of China under Grant 2021YFB3101200, and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515012094. *(Corresponding author: Zhirun Zheng.)*

Zhirun Zheng is with the School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, China (e-mail: zhengzhirun2019@gmail.com).

Zhetao Li, Saiqin Long, and Suiming Guo are with the College of Information Science and Technology, Jinan University, Guangzhou, Guangdong 510632, China (e-mail: liztchina@hotmail.com; xxgcxyxtu@sina.com; guosuiming@email.jnu.edu.cn).

Chao Chen is with the College of Computer Science, Chongqing University, Chongqing 400032, China (e-mail: cschaochen@cqu.edu.cn).

Ke Xu is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100091, China (e-mail: xuke@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TDSC.2025.3532599

I. Introduction

OVER the past decade, we have witnessed a rapid growth of ride-on-demand (RoD) services, such as Uber, Lyft, and Didi. In the typical RoD service, drivers and riders (a.k.a. users) can obtain location-based services, such as dynamic pricing and ride-matching services, by sharing their locations with RoD server. Particularly, the server returns ride-matching result (a.k.a. ride-matching service) and trip fare to users after obtaining their locations, where the fare is calculated by multiplying a dynamic price multiplier (a.k.a. dynamic pricing service) with a fixed normal price based on trip time and distance. Dynamic pricing is a crucial service in RoD that adaptively balances supply (i.e., the number of cars on the road) and demand (i.e., the number of rider requests) by utilizing the price multiplier [1], [2], [3], [4], [5], [6], [7]. For instance, higher price multipliers can entice more drivers to offer rides (i.e., increase supply) and delay requests from riders who are not in rush (i.e., reduce demand). Despite the convenience, RoD services raise serious privacy issues since the server can directly observe users' real locations [8], [9], [10], [11], [12], [13].

Many state-of-the-art privacy-preserving techniques [10], [11], [14], [15], [16], [17], [18], [19], [20] have been proposed to protect users' location privacy through perturbation, and could be broadly classified into two categories: local (or user-centric) privacy-preserving techniques [10], [14], [15], [16] and central privacy-preserving techniques [11], [17], [18]. The central techniques depend on a trusted-third-party to perform perturbation, whereas the local techniques involve the users performing perturbation themselves. In reality, however, finding the trusted-third-party is not easy as it may trade users' privacy for commercial interests and may also be vulnerable to malicious attacks. For example, Facebook's data breach scandal and Didi's excessive collection of users' private information have spread across social media platforms worldwide [21], [22], further eroding trust in third-party and cloud servers. As a result, developing a local privacy-preserving framework for RoD services becomes crucial in protecting users' location privacy.

Nevertheless, the perturbation can potentially affect the supply-demand balance (i.e., the balance between the number of drivers and riders) within the RoD service area, resulting in poor quality of dynamic pricing service. For instance, riders from the central business district (CBD) are obfuscated to residential

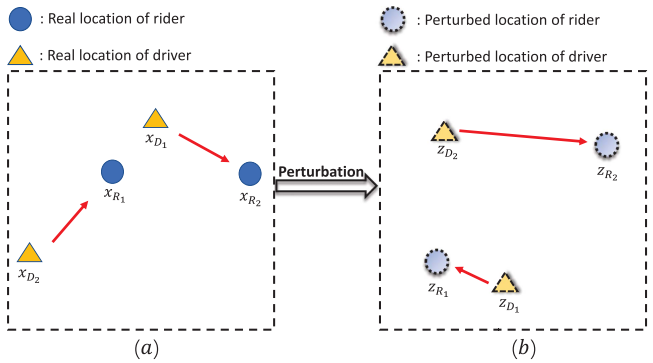


Fig. 1. Motivation example: the ride-matching service is susceptible to the impact of perturbations. (a) The optimal ride-matching result without perturbation is (D_1, R_2) and (D_2, R_1), where $d(x_{D_1}, x_{R_2}) > d(x_{D_1}, x_{R_1})$ and $d(x_{D_2}, x_{R_2}) > d(x_{D_2}, x_{R_1})$. (b) The optimal ride-matching result after perturbation is (D_1, R_1) and (D_2, R_2), where $d(z_{D_1}, z_{R_2}) > d(z_{D_1}, z_{R_1})$ and $d(z_{D_2}, z_{R_2}) > d(z_{D_2}, z_{R_1})$.

areas during evening rush hours, leading to higher pricing in the residential areas compared to the CBD vicinity. As a result, more drivers are attracted to residential areas, despite the fact that there is actually a greater demand for drivers near the CBD during evening rush hours. Furthermore, the ride-matching service is highly sensitive to perturbation as it relies on the distance between drivers and riders [23], [24]. As shown in Fig. 1, the perturbation leads to inaccuracies in the distance, ultimately reducing the quality of ride-matching service.

Different from existing works, we focus on a perturbation-based privacy-preserving framework for the RoD services against an honest-but-curious server without involving any third-party, and aim to provide users with high-quality dynamic pricing and ride-matching services while protecting their location privacy. To this end, we are faced with three challenges. The first challenge is *how to accurately measure the quality loss of dynamic pricing and ride-matching services caused by local perturbation?* Due to perturbation being performed locally, it is not feasible to directly observe the entire location dataset after perturbation. Thus, it is not easy to accurately measure the quality loss of services caused by perturbation in local scenarios. The second challenge is *how to provide users with high-quality dynamic pricing and ride-matching services simultaneously?* Dynamic pricing and ride-sharing services are mutually influential, meaning that preserving the dynamic pricing service may harm the ride-sharing service, and vice versa. Therefore, designing a privacy-preserving framework based on perturbation that offers both high-quality dynamic pricing and ride-matching services is a complex task. The third challenge is *how to achieve a good trade-off between service quality and location privacy?* Achieving high-quality service and location privacy simultaneously is a conflicting goal, and greater perturbation often means better privacy protection but more utility loss. As a result, it is difficult to provide high-quality services while protecting location privacy.

To address the challenges, we propose a pricing-aware differentially private framework for RoD services, called PADP-RoD. Specifically, to measure the quality loss of dynamic pricing and ride-matching services caused by local perturbation, we propose an adaptive supply and demand aware grid to adaptively divide the spatio-temporal region, and then put forward a dynamic pricing utility metric and a ride-matching utility metric based on the grid. After measuring the quality loss, the shortcoming of failing to balance the quality of dynamic pricing and ride-matching services and the location privacy is avoided naturally through structuring a multi-objective minimization problem, whose objective is to minimize the quality loss caused by perturbation. To find the optimal balance between quality and privacy, the multi-objective problem is decomposed into dynamic pricing and ride-matching utility sub-problems, which are solved by the dynamic pricing mapping algorithm and the ride-matching mapping algorithm, respectively. We evaluate PADP-RoD, through a comparison with some state-of-the-art, based on a real-world event-log dataset. The experimental results show that PADP-RoD can provide high-quality dynamic pricing and ride-matching services while defending against the Bayesian inference attack. The main contributions of this paper are summarized as follows:

- We propose a pricing-aware differentially private framework for ride-on-demand services called PADP-RoD. Through applying PADP-RoD, the RoD server can offer high-quality dynamic pricing and ride-matching services without observing users' real locations.
- We propose an adaptive supply and demand aware grid to adapt price multiplier changes. According to the grid, the dynamic pricing and ride-matching utility metrics are proposed to measure the quality loss of services caused by local perturbation.
- Based on the newly proposed metrics, we formulate PADP-RoD as a multi-objective minimization problem. Due to the problem being multi-objective, we relax it into dynamic pricing utility and ride-matching utility sub-problems and propose the dynamic pricing mapping algorithm and the ride-matching mapping algorithm to solve them, respectively. Additionally, we prove that PADP-RoD meets both ε_d-DP and ε_d-identifiability.

The rest of this paper is organized as follows. Section II describes system model, adversary model, and design goals, and then introduces two privacy notions including differential privacy and identifiability. Sections III and IV present the details of PADP-RoD. We propose two utility metrics to measure the utility losses caused by perturbation in Section III. Based on the metrics, in Section IV, we formulate PADP-RoD as a multi-objective optimization problem and propose an efficient algorithm to solve it. We analyze the privacy protection, computational complexity, and limitations of PADP-RoD in Section V. Section VI presents the performance evaluation and results of PADP-RoD. Finally, we review related work in Section VII, and conclude this paper in Section VIII.

II. Models and Preliminaries

A. System Model

As shown in Fig. 2, we consider the pricing-aware privacy-preserving RoD services, and they involve *privacy-conscious*

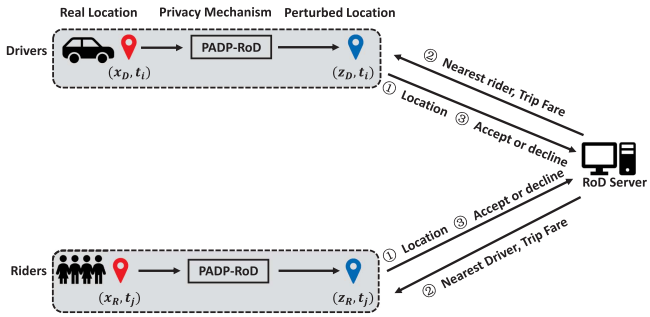


Fig. 2. Workflow of pricing-aware privacy-preserving RoD services.

TABLE I
KEY NOTATIONS AND PARAMETERS

Symbol	Meaning
x_D, y_D, z_D	Real, temporary, or perturbed location of driver
x_R, y_R, z_R	Real, temporary, or perturbed location of rider
$S(*)$, $D(*)$	Supply or demand function
$\rho_s(*)$	Price multiplier elasticity in spatial dimension
$\rho_t(*)$	Price multiplier elasticity in temporal dimension
$Pri(*)$	Dynamic pricing utility metric
$Dis(*)$	Ride-matching utility metric
$\mathcal{B}_{Y\|X}$	Dynamic pricing mapping mechanism
$\mathcal{C}_{Z\|Y}$	Ride-matching mapping mechanism

drivers, *privacy-conscious riders*, and a *semi-honest RoD server*. We consider the server to be semi-honest, that is, the server honestly provides high-quality services for users (i.e., riders and drivers) but is curious about the users' location privacy. Thus, the users perform locally the proposed privacy mechanism, PADP-RoD, to generate perturbed (or false) locations and then share the perturbed locations instead of real locations with the server for location-based services. Particularly, after receiving these perturbed locations, the server returns the trip fare and the ride-matching result (a.k.a. ride-matching service) to users. The trip fare is the product of two items: a dynamic price multiplier that is determined by current supply and demand (a.k.a. dynamic pricing service), and a fixed normal price that depends on trip time and geo-distance. Finally, according to the achieved trip fare and ride-matching result, the users decide whether to accept the services. The key parameters and notations are listed in Table I, and the subscript of locations is removed when distinguishing between the drivers' and riders' locations is unnecessary.

B. Adversary Model

The adversary observes the perturbed locations output by the PADP-RoD, and then launches a Bayesian inference attack [25], [26] to disclose the real locations. Below, we give a detailed definition of the Bayesian inference attack.

Definition 1 (Bayesian Inference Attack): We assume that the adversary's background knowledge is a prior probability distribution $\mathcal{A}_{Z|X}(z \mid x)$ and a victim's profile $p_X(\cdot)$. Given the perturbed location z, the posterior probabilities of its real version can be calculated by Bayes' formula:

$$p(x \mid z) = \frac{\mathcal{A}_{Z|X}(z \mid x) \cdot p_X(x)}{\sum_{x^* \in \mathcal{L}} \mathcal{A}_{Z|X}(z \mid x^*) \cdot p_X(x^*)}, \forall x \in \mathcal{L}. \quad (1)$$

After obtaining the posterior probabilities, the real version of z can be inferred by

$$\widehat{x} = \arg\min_{x \in \mathcal{L}} p(x \mid z). \quad (2)$$

C. Design Goals

Our goal is that the RoD server provides users with high-quality dynamic pricing and ride-matching services while failing to learn their real locations. To balance these conflicting goals, PADP-RoD is formulated as the multi-objective optimization problem:

maximize: Dynamic pricing and ride-matching utility,
subject to: Mapping mechanism $\mathcal{A}_{Z|X}$ satisfies ε-DP.

To obtain the optimal mapping mechanism $\mathcal{A}_{Z|X}$, we need to quantify the dynamic pricing and ride-matching utility losses arising from perturbation and a specialized algorithm to solve the formulated problem, which will be presented in Sections III and IV.

D. Preliminaries

In this part, we detail two different privacy notions including differential privacy [27] and identifiability [28]. Before introducing these privacy notions, we define neighboring locations.

Definition 2 (Neighboring Locations): Let the service area of RoD includes a set of locations $\mathcal{L}$ arranged in M rows and N columns. The location $\forall x \in \mathcal{L}$ is denote as (m, n), where $m \in \{1, \ldots, M\}$ and $n \in \{1, \ldots, N\}$. A pair of locations $x, x' \in \mathcal{L}$ are neighbors when they have a different element. For example, a pair of locations $x := (m, n)$ and $x' := (m, n' \neq n)$ (or $x' := (m' \neq m, n)$) are neighbors, while locations $x := (m, n)$ and $x' := (m' \neq m, n' \neq n)$ are not.

DP is widely used to protect users' location privacy by perturbing their real locations before uploading to the server [15], [16], [29]. When any two neighbor locations have comparable probabilities of perturbing to the same location through a randomized mechanism, the mechanism satisfies DP and can be defined as follows.

Definition 3 (Differential Privacy): Let the service area of RoD includes a set of locations $\mathcal{L}$. The mapping mechanism $\mathcal{A}_{Z|X}$ is said to be ε_d-differential privacy (ε_d-DP) for some $\varepsilon_d \in \mathbb{R}^+$, if for any pair of neighboring locations $\forall x, x' \in \mathcal{L}$:

$$\mathcal{A}_{Z|X}(z \mid x) \leq e^{\varepsilon_d} \cdot \mathcal{A}_{Z|X}(z \mid x'), \forall z \in \mathcal{L}, \quad (3)$$

where $\mathcal{A}_{Z|X}(z \mid x)$ is the probability of perturbing x to z.

Identifiability [28] is defined based on the indistinguishability between any two neighboring locations from a Bayesian view. Particularly, any two neighboring locations in the dataset cannot be distinguished from the posterior probabilities after observing the perturbed location, which makes any users' real location hard to identify. Therefore, identifiability is widely used to defend against Bayesian inference attack [25], [26], and its detailed definition is given below.

Definition 4 (Identifiability): Let the service area of RoD includes a set of locations $\mathcal{L}$. The mapping mechanism $\mathcal{A}_{Z|X}$ satisfies ε_i-identifiability for some $\varepsilon_i \in \mathbb{R}^+$, if for any pair of neighboring locations $x, x' \in \mathcal{L}$:

$$\mathcal{A}_{X|Z}(x \mid z) \leq e^{\varepsilon_i} \cdot \mathcal{A}_{X|Z}(x' \mid z), \forall z \in \mathcal{L}, \tag{4}$$

where $\mathcal{A}_{X|Z}(x \mid z)$ is the posterior probability after observing the perturbed location z.

III. Design of Utility Loss Metrics

A. Adaptive Supply and Demand Aware Grid

The price multiplier is prone to sudden change near hotspots, such as CBD (spatial dimension) and morning peak (temporal dimension) [1], [2]. To adapt to the changes, we propose an adaptive supply and demand aware grid, i.e., placing smaller (or finer) cells near hotspots and dividing other regions (i.e., far from hotspots) into larger (or coarser) cells. In this way, we can guarantee that the price multipliers within the same cell are nearly identical, while those in adjacent cells are distinct. To this end, we propose price multiplier elasticity to measure the rate-of-change of price multiplier in spatial and temporal dimensions, respectively. Then, based on the price multiplier elasticity, we propose an adaptive supply and demand aware discretization algorithm to divide the spatio-temporal region.

The price multiplier is determined by supply-demand relationship, which means that the price multiplier varies as the supply-demand relationship changes. Thus, we define the price multiplier elasticity as the absolute value of the slope of supply-demand relationship, and its detailed definition is given below.

Definition 5 (Price Multiplier Elasticity in Spatial Dimension): Let the supply and demand functions be $S(x,t)$ and $D(x,t)$, respectively. We formulate the supply-demand relationship as $S(x,t) - D(x,t)$. Then, in the spatial dimension, the price multiplier elasticity is defined as

$$\rho_s(x,t) = \left| \frac{\partial S(x,t)}{\partial x} - \frac{\partial D(x,t)}{\partial x} \right|. \tag{5}$$

Definition 6 (Price Multiplier Elasticity in Temporal Dimension): Let $S(x,t)$ is the supply function and $D(x,t)$ is the demand function. The supply-demand relationship is formulated as $S(x,t) - D(x,t)$. Then, in the temporal dimension, the price multiplier elasticity is defined as

$$\rho_t(t) = \left| \frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega} S(x,t) - D(x,t) dx \right|. \tag{6}$$

The price multiplier elasticity $\rho_s(x,t)$ defined in Definition 5 will be greater if the location x is a hotspot during time t. Similarly, if the time t is a hotspot, the price multiplier elasticity $\rho_t(t)$ defined in Definition 6 will be larger. Thus, the fine-grained cells need to be placed near location x during time t when $\rho_s(x,t)$ is larger, and the spatial grid structure needs to be re-adjusted during time t when $\rho_t(t)$ is greater. Next, we detail how to discretize spatio-temporal region based on the price multiplier elasticity.

We adopt the CVDT structure [30] to discretize the spatial region. Compared with the traditional grid structures, such as uniform grid [31] and two-step grid [17], the CVDT grid adjusts the grid size/structure adaptively according to the price multiplier elasticity $\rho_s(x,t)$. Then, we give the definition of CVDT and the commonly used construction algorithm below.

Algorithm 1: Lloyd Algorithm for CVT.

Input : An spatial region Ω, a positive integer g_n, a time t, and a threshold for convergence η
Output: Centroidal Voronoi tessellation $\{l_i, V_i\}_{i=1}^{g_n}$

1 Initialize a set of g_n generators $\{l_i\}_{i=1}^{g_n} \subset \Omega$;
2 **while** *True* **do**
3 Construct the Voronoi tessellation $\{V_i\}_{i=1}^{g_n}$ of Ω according to the generators $\{l_i\}_{i=1}^{g_n}$;
4 Determine the set of centroids, i.e., $\{l_i^*\}_{i=1}^{g_n}$, of Voronoi tessellation $\{V_i\}_{i=1}^{g_n}$ according to the Eq. (8);
5 **if** $\sum_{i=1}^{g_n} d(l_i^*, l_i) \leq \eta$ **then**
6 Update the set of generators: $\{l_i\}_{i=1}^{g_n} \leftarrow \{l_i^*\}_{i=1}^{g_n}$;
7 Break;
8 **end**
9 Update the set of generators: $\{l_i\}_{i=1}^{g_n} \leftarrow \{l_i^*\}_{i=1}^{g_n}$;
10 **end**

Definition 7 (Centroidal Voronoi Delaunay Triangulation): Given a spatial service area Ω and a set of generators $\{l_i\}_{i=1}^{g_n} \subset \Omega$. We define the Voronoi region V_i corresponding to generator l_i as

$$V_i = \{l \in \Omega \mid d(l, l_i) < d(l, l_j), \forall j \in \{1, \ldots, g_n\}/\{i\}\}, \tag{7}$$

where $d(l, *)$ denotes the geo-distance between locations l and $*$. It could be seen from (7), the Voronoi region $V_i \cap V_j = \emptyset$, $\forall j \neq i$ and $\cup_{i=1}^{g_n} \overline{V}_i = \Omega$, i.e., $\{V_i\}_{i=1}^{g_n}$ is a tessellation of Ω. We refer to a Voronoi tessellation $\{(x_i, V_i)\}_{i=1}^{g_n}$ of Ω as a Centroidal Voronoi Tessellation (CVT) if and only if for each generator l_i is the centroid of its corresponding region V_i, i.e.,

$$l_i = \frac{\int_{V_i} l \cdot \rho_s(l,t) dl}{\int_{V_i} \rho_s(l,t) dl}, \ \forall i \in \{1, \ldots, g_n\}. \tag{8}$$

The corresponding dual tessellation is the so-called CVDT.

As detailed in the Definition 7, the generators are pulled around the locations where price multiplier elasticity $\rho_s(x,t)$ is larger (i.e., hotspots in the spatial dimension), resulting in the division into finer cells around the hotspots. The Lloyd algorithm used to construct CVDT is presented in Algorithm 1. We use initialization generators $\{l_i\}_{i=1}^{g_n}$ as the starting point of the next iterator (line 1). At each iteration, we construct the Voronoi tessellation $\{V_i\}_{i=1}^{g_n}$ according to the generators $\{l_i\}_{i=1}^{g_n}$ (line 3), then calculate the centroid of each Voronoi according to (8) (line 4), and finally use these centroids as new generators (line 6). After many iterations, we can obtain CVT of the spatial region (i.e., $\{l_i, V_i\}_{i=1}^{g_n}$) when the centroids of $\{V_i\}_{i=1}^{g_n}$ are no longer moves (line 5). The dual tessellation of $\{l_i, V_i\}_{i=1}^{g_n}$ is the CVDT of spatial region.

We adopt the dichotomy strategy [11] to discretize the temporal region. In each refinement iteration, we mark the cells with larger price multiplier elasticity $\rho_t(t)$, and then subdivide

Algorithm 2: Adaptive Supply and Demand Aware Discretization Algorithm.

Input : A spatial region Ω, a positive integer g_n, a refinement number ref, and a percentage of cells that need to be refined ref_{per}
Output: Adaptive supply and demand aware grid $\mathbb{G}$

1 Initialize uniform point set in the temporal region $[0, 24 \times 60 \text{ minutes}]$, i.e., set T;
2 **for** $i \in [1, ref]$ **do**
3 Calculate the price multiplier elasticity per temporal cell by $\int_{t_i}^{t_{i+1}} \rho_t(t)dt,\ \forall t_i \in T$;
4 Mark top $|ref_{per} \cdot (|T| + 1)|$ temporal cells of price multiplier elasticity;
5 Update the temporal grid T by bisecting each marked cell;
6 **end**
7 **for** $t_i \in T$ **do**
8 $\{l_j, V_j\}_{j=1}^{g_n} \leftarrow$ Discretize Ω by Algorithm 1;
9 $\mathbb{D}_i \leftarrow$ dual tessellation of $\{l_i, V_i\}_{i=1}^{g_n}$;
10 Append $\mathbb{D}_i$ to $\mathbb{G}$;
11 **end**

each marked cell into two equal-sized cells. After the above discussion, we propose an adaptive supply and demand aware discretization algorithm that combines CVDT structure and dichotomy strategy as presented in Algorithm 2. Particularly, we place a uniform grid in the temporal region $[0, 24 \times 60 \text{ minutes}]$ (line 1). At each iteration step in the temporal dimension (lines 2 to 6), we calculate the price multiplier elasticity of each temporal cell by $\int_{t_i}^{t_{i+1}} \rho_t(t)dt$ (temporal cell is defined as $(t_i, t_{i+1}), \forall t_i \in T$) (line 3), then mark the top $ref_{per} \cdot (|T| + 1)$ temporal cells of price multiplier elasticity as needing to be refined (line 4), and finally bisect each marked cell (line 5). After ref iterations, we can obtain an adaptive temporal discretization grid $T = \{t_1, t_2, \ldots\}$. According to the obtained temporal grid T, we re-discretize the spatial region at time $\forall t_i \in T$ by Algorithm 1 (lines 7 to 10), and obtain an adaptive supply and demand aware grid $\mathbb{G} = \{\mathbb{D}_i \mid \forall t_i \in T\}$.

B. Dynamic Pricing Utility Metric

In the local scenario, the utility loss caused by perturbation cannot be measured from the dataset-level viewpoint, because the perturbation is performed locally and the location dataset cannot be completely observed. Thus, we propose the dynamic pricing utility metric based on the grid $\mathbb{G}$ from the user-level perspective to measure the dynamic pricing utility loss arising from local perturbation. Particularly, we can preserve the dynamic pricing utility by maintaining the supply-demand relationship within each cell. As illustrated in Fig. 3, the dynamic pricing utility is maintained when perturbation occurs within the same cell, meaning that the real location and its perturbed version are in the same cell. Conversely, if the perturbation causes the real location to be moved to different cells, the dynamic pricing utility can be undermined. This approach is practical in the local scenario since perturbation occurring within a cell does not alter the supply-demand relationship within that particular cell. After

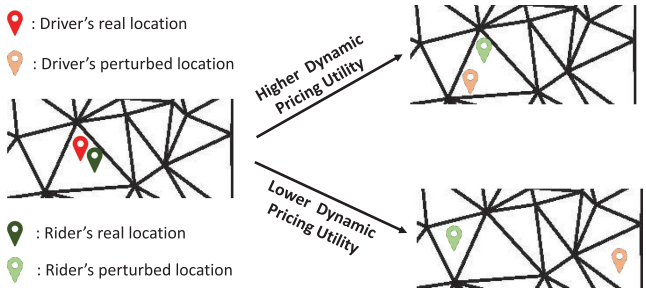


Fig. 3. An intuitive example of measuring the dynamic pricing utility. Perturbation occurring within the same cell results in higher dynamic pricing utility, whereas perturbation that moves real locations to other cells leads to lower dynamic pricing utility. Additionally, when the dynamic pricing utility is preserved, perturbation still leads to inaccurate geo-distance between the driver's location and the rider's location.

the discussion above, we will give a detailed definition of the dynamic pricing utility metric below.

Definition 8 (Dynamic Pricing Utility Metric): Let $\forall x \in \mathcal{L}$ (or $\forall z \in \mathcal{L}$) is the real (or perturbed) location, $\mathcal{A}_{Z|X}(z \mid x)$ is a randomized obfuscation mechanism of perturbing x to z, and $\mathbb{G} = \{\mathbb{D}_i \mid \forall t_i \in T\}$ is the adaptive supply and demand aware grid. Given time (t_i, t_{i+1}), the dynamic pricing utility metric is defined as

$$Pri(\mathcal{A}_{Z|X}) := \sum_{x,z \in \mathcal{L}} p_X(x) \cdot \mathcal{A}_{Z|X}(z \mid x) \cdot \mathbb{1}_{\mathbb{D}_i(z) \neq \mathbb{D}_i(x)}. \quad (9)$$

The indicator function $\mathbb{1}_{\mathbb{D}_i(z) \neq \mathbb{D}_i(x)}$ is formulated as

$$\mathbb{1}_{\mathbb{D}_i(z) \neq \mathbb{D}_i(x)} = \begin{cases} 1 & \text{if } \mathbb{D}_i(z) \neq \mathbb{D}_i(x) \\ 0 & \text{if } \mathbb{D}_i(z) = \mathbb{D}_i(x) \end{cases}, \quad (10)$$

where $\mathbb{D}_i(*)$ returns the cell of location $*$ in the grid $\mathbb{D}_i$ during (t_i, t_{i+1}). For example, $\mathbb{D}_i(*) = C_j^i$ $(C_j^i \in \mathbb{D}_i)$ means that the location $*$ is located in the spatial cell $C_j^i = (m, n)$ $(m \in \{1, \ldots, M\}, n \in \{1, \ldots, N\})$ during (t_i, t_{i+1}).

Definition 8 shows that a larger dynamic pricing utility metric means a more serious utility loss. Therefore, we can preserve dynamic pricing utility by minimizing the dynamic pricing utility metric given the privacy constraint. That said, the RoD server can provide users with high-quality dynamic pricing service without observing the users' real locations.

C. Ride-Matching Utility Metric

The ride-matching service is critical to the RoD service [32], but perturbation leads to worse ride-matching utility. This is because there are notable discrepancies in the geo-distance between perturbed locations when compared to the geo-distance between real locations, i.e., $d(z_R, z_D) \neq d(x_R, x_D)$. Furthermore, as shown in Fig. 3, perturbation occurring within the same cell can maintain dynamic pricing utility, but cannot guarantee that the geo-distance between perturbed locations is approximately equal to the geo-distance between their real locations. As a result, the ride-matching utility cannot sufficiently be preserved only by maintaining dynamic pricing utility.

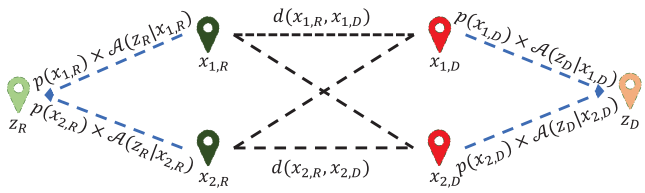


Fig. 4. An intuitive example of calculating the expected travel distance, where $\mathcal{L}$ is formed by two rectangles of equal size (i.e., $|\mathcal{L}| = 2$). The expected travel distance between perturbed locations z_R and z_D (i.e., $d^*(z_R, z_D)$) is the sum of four parts: $z_R \leftarrow x_{1,R} - x_{1,D} \rightarrow z_D$, $z_R \leftarrow x_{1,R} - x_{2,D} \rightarrow z_D$, $z_R \leftarrow x_{2,R} - x_{1,D} \rightarrow z_D$, and $z_R \leftarrow x_{2,R} - x_{2,D} \rightarrow z_D$, where $z_R \leftarrow x_{1,R} - x_{1,D} \rightarrow z_D$ denotes $d(z_R, z_D) = d(x_{1,R}, x_{1,D})$ with probability $p_X(x_{1,R}) \times \mathcal{A}(z_R \mid x_{1,R}) \times p_X(x_{1,D}) \times \mathcal{A}(z_D \mid x_{1,D})$.

Before giving a definition of ride-matching utility metric, we estimate the geo-distance between real locations by *the expected travel distance between perturbed locations*. As shown in Fig. 4, the geo-distance $d(x_R, x_D)$ can be estimated by the expected travel distance $d^*(z_R, z_D)$, where $d^*(z_R, z_D)$ is the sum of four parts: $z_R \leftarrow x_{1,R} - x_{1,D} \rightarrow z_D$, $z_R \leftarrow x_{1,R} - x_{2,D} \rightarrow z_D$, $z_R \leftarrow x_{2,R} - x_{1,D} \rightarrow z_D$, and $z_R \leftarrow x_{2,R} - x_{2,D} \rightarrow z_D$. To illustrate the four parts, we will use $z_R \leftarrow x_{1,R} - x_{1,D} \rightarrow z_D$ as a case. According to the mapping mechanism $\mathcal{A}_{Z|X}$ and the probability $p_X(x_{1,R})$ ($p_X(x_{1,R})$ can be inferred from historical location data), we know that the real version of perturbed location z_R is $x_{1,R}$ with probability $p_X(x_{1,R}) \times \mathcal{A}(z_R \mid x_{1,R})$. Similarly, the real version of perturbed location z_D is $x_{1,D}$ with probability $p_X(x_{1,D}) \times \mathcal{A}(z_D \mid x_{1,D})$. Thus, $d(z_R, z_D) = d(x_{1,R}, x_{1,D})$ with probability $p_X(x_{1,R}) \times p_X(x_{1,D}) \times \mathcal{A}(z_R \mid x_{1,R}) \times \mathcal{A}(z_D \mid x_{1,D})$, that is, $z_R \leftarrow x_{1,R} - x_{1,D} \rightarrow z_D := p_X(x_{1,R}) \times p_X(x_{1,D}) \times \mathcal{A}(z_R \mid x_{1,R}) \times \mathcal{A}(z_D \mid x_{1,D}) \times d(x_{1,R}, x_{1,D})$. Similar for other parts, we have $d^*(z_R, z_D) = \sum_{x_D, x_R \in \mathcal{L}} p_X(x_R) \times p_X(x_D) \times \mathcal{A}(z_R \mid x_R) \times \mathcal{A}(z_D \mid x_D) \times d(x_R, x_D)$. Then, the detailed definition of the expected travel distance is given below.

Definition 9 (Expected Travel Distance): Let $\forall z_D \in \mathcal{L}$ (or $\forall z_R \in \mathcal{L}$) is the driver's (or rider's) perturbed location, and $x_D \in \mathcal{L}$ (or $x_R \in \mathcal{L}$) is the real version of perturbed location z_D (or z_R). Given the mapping mechanism $\mathcal{A}_{Z|X}$ and the profile $p_X(\cdot)$, the expected travel distance between z_D and z_R is defined as

$$d^*(z_D, z_R) = \sum_{x_D, x_R \in \mathcal{L}} \{p_X(x_D) \cdot p_X(x_R) \times \mathcal{A}_{Z|X}(z_D \mid x_D) \cdot \mathcal{A}_{Z|X}(z_R \mid x_R) \cdot d(x_D, x_R)\}, \quad (11)$$

where $d(x_D, x_R)$ is the geo-distance between real locations x_D and x_R.

The expected travel distance $d^*(\cdot)$ can be calculated directly by the RoD server and used instead of $d(\cdot)$ in the ride-matching service. However, relying on the expected travel distance $d^*(z_D, z_R)$ to estimate the geo-distance $d(x_D, x_R)$ is insufficiently accurate, i.e., $|d(x_D, x_R) - d^*(z_D, z_R)| \gg 0$. To this end, we adopt the error $|d(x_D, x_R) - d^*(z_D, z_R)|$ to measure ride-matching utility loss caused by perturbation, and its detailed definition is given below.

Definition 10 (Ride-Matching Utility Metric (RMUM)): Let $\forall x_D \in \mathcal{L}$ (its perturbed version is $z_D \in \mathcal{L}$) is the driver's real location and $\forall x_R \in \mathcal{L}$ (its perturbed version is $z_R \in \mathcal{L}$) is the rider's real location. Then, the ride-matching utility metric is defined as

$$Dis(\mathcal{A}_{Z|X}) := \sum_{x_D, x_R \in \mathcal{L}} |d(x_D, x_R) - d^*(z_D, z_R)|, \quad (12)$$

where $d(\cdot)$ is the geo-distance and the expected travel distance $d^*(\cdot)$ is defined in Definition 9.

Definition 10 shows that a larger ride-matching utility metric means a serious utility loss. Thus, the ride-matching utility can be preserved by minimizing the ride-matching utility metric given the privacy constraint. That said, although the RoD server only observes the perturbed locations, it still provides users with high-quality ride-matching service.

IV. Pricing-Aware Mapping Mechanism

Powered by the proposed dynamic pricing (detailed in Definition 8) and ride-matching (detailed in Definition 10) utility metrics, PADP-RoD is formulated as a muti-object optimization problem ($\mathcal{O}$), minimizing utility losses under the constraint of differential privacy (detailed in Section IV-A). In this way, the RoD server can provide high-quality dynamic pricing and ride-matching services while protecting location data privacy. To this end, we solve the problem ($\mathcal{O}$) in Section IV-B, finding the optimal privacy mapping mechanism. Specifically, as shown in Fig. 5, we obtain high-quality dynamic pricing service by ensuring perturbation occurs within the same cell (i.e., the privacy mapping mechanism $\mathcal{B}_{Y|X}$), and then achieve high-quality ride-matching service while maintaining the quality of dynamic pricing service by adjusting the perturbed locations within the same cells (i.e., the privacy mapping mechanism $\mathcal{C}_{Z|Y}$). With this idea, the multi-objective optimization problem ($\mathcal{O}$) over the mapping mechanism $\mathcal{A}_{Z|X}$ is relaxed to two single-objective optimization problems ($\mathcal{O}_d$) (which over the mapping mechanism $\mathcal{B}_{Y|X}$) and ($\mathcal{O}_r$) (which over the mapping mechanism $\mathcal{C}_{Z|Y}$), respectively. Finally, as detailed in Sections IV-B1 and IV-B2, we employ the proposed dynamic pricing mapping algorithm and ride-matching mapping algorithm to solve problems ($\mathcal{O}_d$) and ($\mathcal{O}_r$), respectively.

A. Problem Formulation

Intuitively, the better privacy the users need, the less utility they receive, and vice versa. Thus, we formulate the muti-objective optimization problem given in Section II-C as

$$(\mathcal{O}): \underset{\mathcal{A}_{Z|X}(z|x)}{\text{minimize}} \ \left(Pri(\mathcal{A}_{Z|X}), Dis(\mathcal{A}_{Z|X})\right), \quad (13a)$$

$$\text{subject to} \ \begin{array}{l} \mathcal{A}_{Z|X}(z \mid x) \le e^{\varepsilon_d} \cdot \mathcal{A}_{Z|X}(z \mid x'), \\ \forall x, x' : x \sim x', z \in \mathcal{L}, \end{array} \quad (13b)$$

$$\sum_{z \in \mathcal{L}} \mathcal{A}_{Z|X}(z \mid x) = 1, \ \forall x \in \mathcal{L}, \quad (13c)$$

$$\mathcal{A}_{Z|X}(z \mid x) \ge 0, \ \forall x, z \in \mathcal{L}, \quad (13d)$$

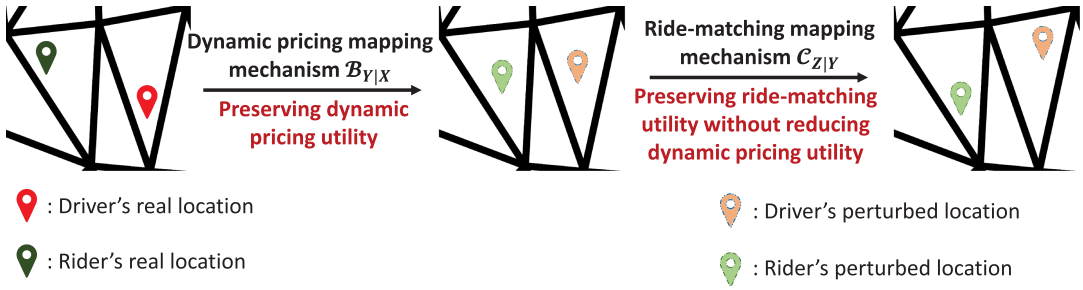


Fig. 5. The workflow of PADP-RoD. The dynamic pricing mapping mechanism $\mathcal{B}_{Y|X}$ ensures that perturbation occurs in the same cell (i.e., the driver's real location and its perturbed version are in the same cell), and the ride-matching mapping mechanism $\mathcal{C}_{Z|Y}$ only adjusts the users' locations within the same cell. In this way, we can preserve both the dynamic pricing utility and the ride-matching utility.

Algorithm 3: Dynamic Pricing Mapping Algorithm.

Input : θ_1: Threshold for convergence, $p_X(x)$: Probability of real location
Output: $\mathcal{B}_{Y|X}(y \mid x)$: Dynamic pricing mapping mechanism

Initialize the dynamic pricing mapping mechanism $\mathcal{B}^0_{Y|X}(y \mid x)$ as a uniform distribution, i.e.,

$$\mathcal{B}^0_{Y|X}(y \mid x) = \begin{pmatrix} 1/|\mathcal{L}| & \cdots & 1/|\mathcal{L}| \\ \vdots & \ddots & \vdots \\ 1/|\mathcal{L}| & \cdots & 1/|\mathcal{L}| \end{pmatrix}_{|\mathcal{L}|*|\mathcal{L}|};$$

Calculate $p_Y(y)$ using $\mathcal{B}^0_{Y|X}(y \mid x)$ by Eq. (23);
$k \leftarrow 1$;
while *True* **do**
  Calculate $\mathcal{B}^k_{Y|X}(y \mid x)$ using $p_Y(y)$ by Eq. (21);
  Calculate $p_Y(y)$ using $\mathcal{B}^k_{Y|X}(y \mid x)$ by Eq. (23);
  Normalize $\mathcal{B}^k_{Y|X}(y \mid x)$ by Eq. (24);
  if $\|\mathcal{B}^k_{Y|X}(y \mid x) - \mathcal{B}^{k-1}_{Y|X}(y \mid x)\|_p \leq \theta_1$ **then**
    $\mathcal{B}_{Y|X}(y \mid x) \leftarrow \mathcal{B}^k_{Y|X}(y \mid x)$;
    Break;
  end
  $k \leftarrow k + 1$;
end

where ε_d represents the privacy budget, and the mapping mechanism $\mathcal{A}_{Z|X}$ is $|\mathcal{L}| \times |\mathcal{L}|$-dimension matrix. Then, we can obtain an optimal local mapping mechanism $\mathcal{A}_{Z|X}(z \mid x)$ by solving the problem $(\mathcal{O})$. Particularly, PADP-RoD works in a *user-centric* manner, i.e., PADP-RoD perturbs the real location x to the perturbed location z *locally* according to the mapping mechanism $\mathcal{A}_{Z|X}(z \mid x)$.

B. Optimizing the Formulated Problem

The problem $(\mathcal{O})$ is a multi-objective optimization, thus solving it directly and efficiently is prohibitive. With this concern, we relax the problem $(\mathcal{O})$ into two single-objective optimizations and then solve them respectively. Particularly, we break the mapping mechanism $\mathcal{A}_{Z|X}(z \mid x)$ into two concatenated stages by

$$\mathcal{A}_{Z|X}(z \mid x) = \sum_{y \in \mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) \cdot \mathcal{B}_{Y|X}(y \mid x), \tag{14}$$

where $\mathcal{B}_{Y|X}(y \mid x)$ is the dynamic pricing mapping mechanism and $\mathcal{C}_{Z|Y}(z \mid y)$ denotes the ride-matching mapping mechanism. As shown in Fig. 5, we preserve dynamic pricing utility by the mapping mechanism $\mathcal{B}_{Y|X}$ (perturbation occurs within the same cell), and then preserve the ride-matching utility while maintaining dynamic pricing utility by the mapping mechanism $\mathcal{C}_{Z|Y}$ (adjust the users' locations within the same cell). According to this idea, we relax the problem $(\mathcal{O})$ into a dynamic pricing utility sub-problem $(\mathcal{O}_d)$ and a ride-matching utility sub-problem $(\mathcal{O}_r)$, where the optimization over $\mathcal{B}_{Y|X}$ is sub-problem $(\mathcal{O}_d)$ and the optimization over $\mathcal{C}_{Z|Y}$ given $\mathcal{B}_{Y|X}$ is sub-problem $(\mathcal{O}_r)$.

The mapping mechanisms $\mathcal{B}_{Y|X}$ and $\mathcal{C}_{Z|Y}$ are indispensable to preserve both dynamic pricing and ride-matching utility. We can obtain minimal dynamic pricing utility loss, i.e., $\arg\min Pri(\mathcal{B}_{Y|X})$, by solving the sub-problem $(\mathcal{O}_d)$. Then, we take $pri(\mathcal{C}_{Z|Y}) \leq \arg\min Pri(\mathcal{B}_{Y|X}) + \Phi$ as a constraint of sub-problem $(\mathcal{O}_r)$ to maintain dynamic pricing utility, where Φ is the upper bound of dynamic pricing utility loss caused by $\mathcal{C}_{Z|Y}$. When nothing is known about $\arg\min Pri(\mathcal{B}_{Y|X})$, we cannot give an appropriate dynamic pricing utility loss constraint of sub-problem $(\mathcal{O}_r)$.

1) Dynamic Pricing Utility Sub-Problem: The dynamic pricing utility sub-problem is formulated as,

$$(\mathcal{O}_d): \underset{\mathcal{B}_{Y|X}(y|x)}{\text{minimize}} \sum_{x,y \in \mathcal{L}} p_X(x) \cdot \mathcal{B}_{Y|X}(y \mid x) \cdot \mathbb{1}_{\mathbb{D}_i(y) \neq \mathbb{D}_i(x)}, \tag{15a}$$

$$\text{subject to } \mathcal{B}_{Y|X}(y \mid x) \leq e^{\varepsilon_d} \cdot \mathcal{B}_{Y|X}(y \mid x'), \quad \forall x, x' : x \sim x', y \in \mathcal{L}, \tag{15b}$$

$$\sum_{y \in \mathcal{L}} \mathcal{B}_{Y|X}(y \mid x) = 1, \ \forall x \in \mathcal{L}, \tag{15c}$$

$$\mathcal{B}_{Y|X}(y \mid x) \geq 0, \ \forall x, y \in \mathcal{L}, \tag{15d}$$

where the mapping mechanism $\mathcal{B}_{Y|X}$ is $|\mathcal{L}| \times |\mathcal{L}|$-dimension matrix.

Since $p_X(x) \cdot \mathcal{B}_{Y|X}(y \mid x) = \mathcal{B}_{X,Y}(x, y) = p_Y(y) \cdot \mathcal{B}_{X|Y}(x \mid y)$, we have

$$\sum_{x,y \in \mathcal{L}} p_X(x) \cdot \mathcal{B}_{Y|X}(y \mid x) \cdot \mathbb{1}_{\mathbb{D}_i(y) \neq \mathbb{D}_i(x)} = \sum_{x,y \in \mathcal{L}} p_Y(y) \cdot \mathcal{B}_{X|Y}(x \mid y) \cdot \mathbb{1}_{\mathbb{D}_i(y) \neq \mathbb{D}_i(x)}. \tag{16}$$

We know from [33, Theorem 1] that DP and identifiability (defined in Definition 4) imply each other, i.e., if the mapping mechanism $\mathcal{B}_{Y|X}$ satisfies ε_d-DP then it also satisfies ε_d-identifiability; similarly if the mapping mechanism $\mathcal{B}_{Y|X}$ satisfies ε_d-identifiability then it also satisfies ε_d-DP. Therefore, the constraint of ε_d-DP, i.e., (15b), could be equivalently replaced by the constraint of ε_d-identifiability, i.e.,

$$\mathcal{B}_{X|Y}(x \mid y) \leq e^{\varepsilon_d} \cdot \mathcal{B}_{X|Y}(x' \mid y),\ \forall x, x' : x \sim x', y \in \mathcal{L}. \tag{17}$$

In summary, we convert the sub-problem $(\mathcal{O}_d)$ to its equivalent form $(\mathcal{O}_d^1)$, i.e.,

$$(\mathcal{O}_d^1):\ \underset{\mathcal{B}_{X|Y}, p_Y}{\text{minimize}} \sum_{x,y\in\mathcal{L}} p_Y(y) \cdot \mathcal{B}_{X|Y}(x \mid y) \cdot \mathbb{1}_{\mathbb{D}_i(y)\neq\mathbb{D}_i(x)}, \tag{18a}$$

$$\text{subject to}\ \begin{matrix}\mathcal{B}_{X|Y}(x \mid y) \leq e^{\varepsilon_d} \cdot \mathcal{B}_{X|Y}(x' \mid y),\\ \forall x, x' : x \sim x', y \in \mathcal{L},\end{matrix} \tag{18b}$$

$$\sum_{x\in\mathcal{L}} \mathcal{B}_{X|Y}(x \mid y) = 1,\ \forall y \in \mathcal{L}, \tag{18c}$$

$$\mathcal{B}_{X|Y}(x \mid y) \geq 0,\ \forall x, y \in \mathcal{L}, \tag{18d}$$

$$\sum_{y\in\mathcal{L}} p_Y(y) = 1, \tag{18e}$$

$$p_Y(y) \geq 0,\ \forall y \in \mathcal{L}. \tag{18f}$$

From [28, Lemma 1], we know that the optimal solution of problem $(\mathcal{O}_d^1)$ for any fixed $y \in \mathcal{L}$ is

$$\mathcal{B}_{X|Y}(x \mid y) = \frac{e^{-\varepsilon_d \cdot \mathbb{1}_{\mathbb{D}_i(y)\neq\mathbb{D}_i(x)}}}{\left(1 + (|\mathcal{L}| - 1) \cdot e^{-\varepsilon_d}\right)^2},\ \forall x \in \mathcal{L}. \tag{19}$$

Eq. (19) shows that we assign larger probability to $\mathcal{B}_{X|Y}(x \mid y)$ with $\mathbb{D}_i(x) = \mathbb{D}_i(y)$, and smaller probability to $\mathcal{B}_{X|Y}(x \mid y)$ with $\mathbb{D}_i(x) \neq \mathbb{D}_i(y)$. Due to the fact that

$$\mathcal{B}_{X|Y}(x \mid y) = \frac{\mathcal{B}_{X,Y}(x, y)}{p_Y(y)} = \frac{\mathcal{B}_{Y|X}(y \mid x) \cdot p_X(x)}{p_Y(y)}, \tag{20}$$

we can obtain the optimal solution of sub-problem $(\mathcal{O}_d)$, i.e.,

$$\mathcal{B}_{Y|X}(y \mid x) = \frac{p_Y(y) \cdot e^{-\varepsilon_d \cdot \mathbb{1}_{\mathbb{D}_i(y)\neq\mathbb{D}_i(x)}}}{p_X(x) \cdot \left(1 + (|\mathcal{L}| - 1) \cdot e^{-\varepsilon_d}\right)^2},\ \forall x, y \in \mathcal{L}, \tag{21}$$

where the probability $p_Y(y)$ is unknown.

According to the above discussion, to obtain the mapping mechanism $\mathcal{B}_{Y|X}$, we propose a dynamic pricing mapping algorithm as presented in Algorithm 3. We start with an initial mechanism, i.e., the uniform mapping mechanism

$$\mathcal{B}_{Y|X}(y \mid x) = \begin{pmatrix} 1/|\mathcal{L}| & \cdots & 1/|\mathcal{L}| \\ \vdots & \ddots & \vdots \\ 1/|\mathcal{L}| & \cdots & 1/|\mathcal{L}| \end{pmatrix}_{|\mathcal{L}|*|\mathcal{L}|}. \tag{22}$$

Then, we iterate the following three steps in order:

Algorithm 4: Ride-Matching Mapping Algorithm.

Input : θ_2: Threshold for convergence, $\delta > 0$: Penalty factor, $\rho > 1$: Penalty factor growth coefficient, S: Swarm size, Φ: Upper bound of pricing utility loss

Output: $\mathcal{C}_{Z|Y}(z \mid y)$: Ride-matching mapping mechanism

for *each particle* $k \in [1, S]$ **do**
  Generate the position $\mathcal{C}_{Z|Y}^{0,k}$ and the velocity $V^{0,k}$ randomly;
  Normalize $\mathcal{C}_{Z|Y}^{0,k}$ by Eq. (33);
end
$Gbest^0 \leftarrow \mathcal{C}_{Z|Y}^{0,1}$;
for *each particle* $k \in [1, S]$ **do**
  if $P(\mathcal{C}_{Z|Y}^{0,k}, \delta) < P(Gbest^0, \delta)$ **then**
    $Gbest^0 \leftarrow \mathcal{C}_{Z|Y}^{0,k}$;
  end
end
$Gbest^1, \{\mathcal{C}_{Z|Y}^{1,1}, \cdots, \mathcal{C}_{Z|Y}^{1,S}\}, \{V^{1,1}, \cdots, V^{1,S}\} \leftarrow \min_{\mathcal{C}_{Z|Y}} P(\mathcal{C}_{Z|Y}, \delta)$ is solved by Algorithm 5 with $\{\mathcal{C}_{Z|Y}^{0,1}, \cdots, \mathcal{C}_{Z|Y}^{0,S}\}$ and $\{V^{0,1}, \cdots V^{0,S}\}$;
$\mathcal{C}_{Z|Y}(z \mid y) \leftarrow Gbest^1$;
$j \leftarrow 1$;
while $\|Gbest^{j+1} - Gbest^j\|_p > \theta_2$ **do**
  $\delta \leftarrow \rho \cdot \delta$;
  $Gbest^{j+1}, \{\mathcal{C}_{Z|Y}^{j+1,1}, \cdots, \mathcal{C}_{Z|Y}^{j+1,S}\}, \{V^{j+1,1}, \cdots, V^{j+1,S}\} \leftarrow \min_{\mathcal{C}_{Z|Y}} P(\mathcal{C}_{Z|Y}, \delta)$ is solved by Algorithm 5 with $\{\mathcal{C}_{Z|Y}^{j,1}, \cdots \mathcal{C}_{Z|Y}^{j,S}\}$ and $\{V^{j,1}, \cdots V^{j,S}\}$;
  $\mathcal{C}_{Z|Y}(z \mid y) \leftarrow Gbest^{j+1}$;
  $j \leftarrow j + 1$;
end

1) Fixing the mapping mechanism $\mathcal{B}_{Y|X}$, the probability $p_Y(y)$ is calculated by

$$p_Y(y) = \sum_{x\in\mathcal{L}} p_X(x) \cdot \mathcal{B}_{Y|X}(y \mid x),\ \forall y \in \mathcal{L}. \tag{23}$$

2) After obtaining the probability $p_Y(y)$, the mapping mechanism $\mathcal{B}_{Y|X}(y \mid x)$ is updated by (21).
3) We normalize the mapping mechanism $\mathcal{B}_{Y|X}(y \mid x)$ by

$$\mathcal{B}_{Y|X}(y \mid x) = \frac{\mathcal{B}_{Y|X}(y \mid x)}{\sum_{y^*\in\mathcal{L}} \mathcal{B}_{Y|X}(y^* \mid x)},\ \forall x, y \in \mathcal{L}. \tag{24}$$

Through iterating these steps, the (15a) will be minimized and so does the dynamic pricing utility loss.

2) Ride-Matching Utility Sub-Problem: The ride-matching utility sub-problem $(\mathcal{O}_r)$ is formulated as,

$$(\mathcal{O}_r):\ \underset{\mathcal{C}_{Z|Y}(z|y)}{\text{minimize}} \sum_{y_D, y_R\in\mathcal{L}} \left|d(y_D, y_R) - d^*(z_D, z_R)\right|, \tag{25a}$$

$$\text{subject to}\ \begin{matrix}\mathcal{C}_{Z|Y}(z \mid y) \leq e^{\varepsilon_d/2} \cdot \mathcal{C}_{Z|Y}(z \mid y'),\\ \forall y, y' : y \sim y', z \in \mathcal{L},\end{matrix} \tag{25b}$$

Algorithm 5: PSO for Solving $\min_{\mathcal{C}_{Z|Y}} P(\mathcal{C}_{Z|Y}, \delta)$.

Input : $\{\mathcal{C}_{Z|Y}^{0,1}, \cdots, \mathcal{C}_{Z|Y}^{0,S}\}$: Positions of particle swarm, $\{V^{0,1}, \cdots, V^{0,S}\}$: Velocities of particle swarm, MNI: Maximum number of iterations, d_1, d_2: Cognitive and social acceleration coefficients

Output: $Gbest$: Best-known position of the swarm, $\{Pbest^1, \cdots, Pbest^S\}$: Best-known position of each particle, $\{V^{MNI,1}, \cdots, V^{MNI,S}\}$: Velocity of each particle

1 **for** *each particle* $k \in [1, S]$ **do**
2 $\quad Pbest^{0,k} \leftarrow \mathcal{C}_{Z|Y}^{0,k}$;
3 **end**
4 $Gbest \leftarrow Pbest^{0,1}$;
5 **for** *each particle* $k \in [1, S]$ **do**
6 $\quad$ **if** $P(Pbest^{0,k}, \delta) < P(Gbest, \delta)$ **then**
7 $\quad\quad Gbest \leftarrow Pbest^{0,k}$;
8 $\quad$ **end**
9 **end**
10 $j \leftarrow 0$;
11 **while** $j \leq MNI - 1$ **do**
12 $\quad$ **for** *each particle* $k \in [1, S]$ **do**
13 $\quad\quad$ Update the velocity $V^{j+1,k}$ by Eq. (35);
14 $\quad\quad$ Update the position $\mathcal{C}_{Z|Y}^{j+1,k}$ by Eq. (36);
15 $\quad\quad$ **if** $P(\mathcal{C}_{Z|Y}^{j+1,k}, \delta) < P(Pbest^k, \delta)$ **then**
16 $\quad\quad\quad Pbest^k \leftarrow \mathcal{C}_{Z|Y}^{j+1,k}$;
17 $\quad\quad$ **end**
18 $\quad\quad$ **if** $P(Pbest^k, \delta) < P(Gbest, \delta)$ **then**
19 $\quad\quad\quad Gbest \leftarrow Pbest^k$;
20 $\quad\quad$ **end**
21 $\quad$ **end**
22 $\quad j \leftarrow j + 1$;
23 **end**

$$Pri(\mathcal{C}_{Z|Y}) \leq \arg\min Pri(\mathcal{B}_{Y|X}) + \Phi, \tag{25c}$$

$$\sum_{z \in \mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) = 1, \forall y \in \mathcal{L}, \tag{25d}$$

$$\mathcal{C}_{Z|Y}(z \mid y) \geq 0, \forall y, z \in \mathcal{L}, \tag{25e}$$

where the mapping mechanism $\mathcal{C}_{Z|Y}$ is $|\mathcal{L}| \times |\mathcal{L}|$-dimension matrix.

Given the ride-utility mapping mechanism $\mathcal{C}_{Z|Y}$ and the profile $p_Y(y)$, we re-formulate the expected travel distance $d^*(z_D, z_R)$ (defined in Definition 9) as

$$d^*(z_D, z_R) = \sum_{y_D, y_R \in \mathcal{L}} \{p_Y(y_D) \cdot p_Y(y_R) \times \mathcal{C}_{Z|Y}(z_D \mid y_D) \cdot \mathcal{C}_{Z|Y}(z_R \mid y_R) \cdot d(y_D, y_R)\}. \tag{26}$$

We limit the dynamic pricing utility loss arising from the mapping mechanism $\mathcal{C}_{Z|Y}$ by constraint (25c), where $\arg\min Pri(\mathcal{B}_{Y|X})$ represents the minimal dynamic pricing utility loss caused by $\mathcal{B}_{Y|X}$ (obtain by solving the sub-problem $(\mathcal{O}_d)$) and Φ is the upper bound of dynamic pricing utility caused by $\mathcal{C}_{Z|Y}$.

We employ the *exact penalty function method* [34] to transform the sub-problem $(\mathcal{O}_r)$ into an unconstrained optimization, because $(\mathcal{O}_r)$ is semi-infinite programming [35] and hard to solve directly. With the proper selection of the penalty factor, the optimal solution of the exact penalty function converges to the optimal solution of the original problem. Next, we give a detailed definition of the l_1-penalty function, a commonly used exact penalty function, of sub-problem $(\mathcal{O}_r)$.

Definition 11 (l_1-penalty function): The l_1-penalty function of problem $(\mathcal{O}_r)$ is formulated as

$$P(\mathcal{C}_{Z|Y}(z \mid y), \delta) = \sum_{y_D, y_R \in \mathcal{L}} |d(y_D, y_R) - d^*(z_D, z_R)| + \delta \cdot \left\{ \sum_{y \in \mathcal{L}} |\mathcal{H}(y)| + \sum_{y, y': y \sim y', z \in \mathcal{L}} \mathcal{G}(y, y', z) + \sum_{y, z \in \mathcal{L}} \mathcal{I}(y, z) + \mathcal{K}(\mathcal{C}_{Z|Y}, \Phi) \right\}, \tag{27}$$

where the constant $\delta > 0$ is the penalty factor, and the penalty terms $\mathcal{H}(y), \mathcal{G}(y, y', z), \mathcal{I}(y, z)$ are formulated as

$$\mathcal{H}(y) = \sum_{z \in \mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) - 1, \tag{28}$$

$$\mathcal{G}(y, y', z) = \max\left\{C_{Z|Y}(z \mid y) - e^{\varepsilon_d} \cdot C_{Z|Y}(z \mid y'), 0\right\}, \tag{29}$$

$$\mathcal{I}(y, z) = \max\left\{-\mathcal{C}_{Z|Y}(z \mid y), 0\right\}, \tag{30}$$

$$\mathcal{K}\left(\mathcal{C}_{Z|Y}, \Phi\right) = \max\left\{Pri(\mathcal{C}_{Z|Y}) - \arg\min Pri\left(\mathcal{B}_{Y|X}\right) - \Phi, 0\right\}. \tag{31}$$

We can obtain the mapping mechanism $\mathcal{C}_{Z|Y}$ by solving optimization problem $\min_{\mathcal{C}_{Z|Y}} P(\mathcal{C}_{Z|Y}, \delta)$ when given a suitable penalty factor δ. However, gradient-based optimization algorithms cannot be used to solve the problem $\min_{\mathcal{C}_{Z|Y}} P(\mathcal{C}_{Z|Y}, \delta)$ because the function $P(\mathcal{C}_{Z|Y}, \delta)$ is non-differentiable caused by (29)–(31). Therefore, we adopt Particle Swarm Optimization (PSO) [36], a popular swarm-based algorithm, to solve the problem $\min_{\mathcal{C}_{Z|Y}} P(\mathcal{C}_{Z|Y}, \delta)$. After the above discussion, we propose the ride-matching mapping algorithm that combines the l_1-penalty function method and PSO to solve the problem $(\mathcal{O}_r)$.

Referring to Algorithm 4, we present the details of the ride-matching mapping algorithm. We randomly initialize S particles (lines 1 to 4), which fly in a $|\mathcal{L}| \times |\mathcal{L}|$-dimension search space to seek an optimal solution. Particularly, the 0-th swarm contains S particles, where each particle has a velocity matrix $V^{0,k}, \forall k \in [1, S]$ (velocity of k-th particle in the 0-th swarm) and a position matrix $\mathcal{C}_{Z|Y}^{0,k}, \forall k \in [1, S]$ (position of k-th particle in the 0-th swarm). Each position $\mathcal{C}_{Z|Y}^{0,k}$, a solution candidate of the problem

$(\mathcal{O}_r)$, is initialized as

$$\mathcal{C}_{Z|Y}^{0,k} = \begin{pmatrix} c_{1,1}^{0,k} & \cdots & c_{1,|\mathcal{L}|}^{0,k} \\ \vdots & \ddots & \vdots \\ c_{|\mathcal{L}|,1}^{0,k} & \cdots & c_{|\mathcal{L}|,|\mathcal{L}|}^{0,k} \end{pmatrix}_{|\mathcal{L}|\times|\mathcal{L}|,} \quad \forall k \in [1, S], \tag{32}$$

where $\forall c_{m,n}^{0,k} \in \mathcal{C}_{Z|Y}^{0,k}$ drawn from uniform distribution $U(0, 1)$. Then, $\mathcal{C}_{Z|Y}^{0,k}$ is normalized by

$$c_{m,n}^{0,k} \leftarrow \frac{c_{m,n}^{0,k}}{\sum_{\forall n\in[1,|\mathcal{L}|]} c_{m,n}^{0,k}}, \ \forall m, n \in [1, |\mathcal{L}|]. \tag{33}$$

Similarly, velocity $V^{0,k}$ is the moving velocity of position $\mathcal{C}_{Z|Y}^{0,k}$ and is initialized as

$$V^{0,k} = \begin{pmatrix} v_{1,1}^{0,k} & \cdots & v_{1,|\mathcal{L}|}^{0,k} \\ \vdots & \ddots & \vdots \\ v_{|\mathcal{L}|,1}^{0,k} & \cdots & v_{|\mathcal{L}|,|\mathcal{L}|}^{0,k} \end{pmatrix}_{|\mathcal{L}|\times|\mathcal{L}|,} \quad \forall k \in [1, S], \tag{34}$$

where $\forall v_{m,n}^{0,k} \in V^{0,k}$ drawn from uniform distribution $U(0, 0.1)$. Additionally, we also find the best position $Gbest^0$ from 0-th swarm, i.e., the optimal solution of problem $(\mathcal{O}_r)$ in the 0-th swarm (lines 5 to 10). At each step in the iteration, we increase the penalty coefficient δ according to $\delta = \rho \cdot \delta$ (line 15), then solve the problem $\min_{\mathcal{C}_{Z|Y}} P(\mathcal{C}_{Z|Y}, \delta)$ by PSO given the current swarm (i.e., $\{\mathcal{C}_{Z|Y}^{j,1}, \ldots, \mathcal{C}_{Z|Y}^{j,S}\}$ and $\{V^{j,1}, \ldots, V^{j,S}\}$) as shown in Algorithm 5 (line 16), and finally assign the best position $Gbest^{j+1}$ (the optimal solution candidate of problem $(\mathcal{O}_r)$ in the $j + 1$-th swarm) returned by Algorithm 5 to the mapping mechanism $\mathcal{C}_{Z|Y}$ (line 17). After iterating these steps until convergence (line 14), we obtain the optimal ride-matching mapping mechanism $\mathcal{C}_{Z|Y}$ and the ride-matching utility loss will be minimized.

When the penalty factor δ is fixed, the problem $\min_{\mathcal{C}_{Z|Y}} P(\mathcal{C}_{Z|Y}, \delta)$ can be solved by PSO as presented in Algorithm 5. Given an input swarm $\{\mathcal{C}_{Z|Y}^{0,1}, \ldots, \mathcal{C}_{Z|Y}^{0,S}\}$ and $\{V^{0,1}, \ldots, V^{0,S}\}$, we initialize the best-known position of each particle (the optimal solution candidate of each particle) to its initial position (lines 1 to 3), i.e., $Pbest^k$ (the best-known position of k-th particle), and find the best-known position of the swarm (lines 4 to 9), i.e., $Gbest$ (the optimal solution candidate in the swarm). At each step in the iteration, we update the position and velocity of each particle (lines 13 to 14) by

$$v_{m,n}^{j+1,k} = v_{m,n}^{j,k} + d_1 \cdot r_1 \cdot (pbest_{m,n}^k - c_{m,n}^{j,k}) + d_2 \cdot r_2 \times \left(gbest_{m,n}^k - c_{m,n}^{j,k}\right), \ \forall m, n \in [1, |\mathcal{L}|], \tag{35}$$

$$c_{m,n}^{j+1,k} = c_{m,n}^{j,k} + v_{m,n}^{j+1,k}, \ \forall m, n \in [1, |\mathcal{L}|], \tag{36}$$

where r_1, r_2 are drawn from the uniform distribution $U(0, 1)$, and the subscripts of $pbest_{m,n}^k \in Pbest^k, gbest_{m,n}^k \in Gbest^k$ are defined similarly to $c_{m,n}^{j,k} \in \mathcal{C}_{Z|Y}^{j,k}$. Next, we update the best-known position of each particle (lines 15 to 17), i.e., $Pbest^k, \forall k \in [1, S]$, and then update the best-known position of the swarm (lines 18 to 20), i.e., $Gbest$. Through iterating these steps, the l_1-penalty function will be minimized and return the optimal swarm, i.e., $Gbest$, $\{Pbest^1, \ldots, Pbest^S\}$, $\{V^{MNI,1}, \ldots, V^{MNI,S}\}$.

V. Theoretical Analysis

A. Privacy Analysis

In this part, we prove that PADP-RoD satisfies ε_d-DP and ε_d-identifiability, and their detailed theorems are given below.

Theorem 1: Let $\varepsilon_d \geq 0$. Given the dynamic pricing mapping mechanism $\mathcal{B}_{Y|X}$ satisfies ε_d-DP. We have PADP-RoD satisfies ε_d-DP for any ride-matching mapping mechanism $\mathcal{C}_{Z|Y}$.

Proof: Because $\mathcal{A}_{Z|X}(z \mid x) = \sum_{y\in\mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) \cdot \mathcal{B}_{Y|X}(y \mid x)$, we have

$$\frac{\mathcal{A}_{Z|X}(z \mid x)}{\mathcal{A}_{Z|X}(z \mid x')} = \frac{\sum_{y\in\mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) \cdot \mathcal{B}_{Y|X}(y \mid x)}{\sum_{y\in\mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) \cdot \mathcal{B}_{Y|X}(y \mid x')}, \tag{37}$$

where locations x and x' are neighbors. Since the mechanism $\mathcal{B}_{Y|X}$ satisfies ε_d-DP, we have

$$\frac{\mathcal{B}_{Y|X}(y \mid x)}{\mathcal{B}_{Y|X}(y \mid x')} \leq e^{\varepsilon_d}, \ \forall x, x' : x \sim x', y \in \mathcal{L}. \tag{38}$$

Thus, we have

$$\frac{\sum_{y\in\mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) \cdot \mathcal{B}_{Y|X}(y \mid x)}{\sum_{y\in\mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) \cdot \mathcal{B}_{Y|X}(y \mid x')} \tag{39}$$

$$\leq \frac{\sum_{y\in\mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) \cdot e^{\varepsilon_d} \cdot \mathcal{B}_{Y|X}(y \mid x')}{\sum_{y\in\mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) \cdot \mathcal{B}_{Y|X}(y \mid x')}, \tag{40}$$

$$= e^{\varepsilon_d}. \tag{41}$$

In summary, we have

$$\frac{\mathcal{A}_{Z|X}(z \mid x)}{\mathcal{A}_{Z|X}(z \mid x')} \leq e^{\varepsilon_d}, \ \forall x, x' : x \sim x', z \in \mathcal{L}. \tag{42}$$

That is, the local mapping mechanism $\mathcal{A}_{Z|X}$ satisfies ε_d-DP. □

Theorem 2: Let $\varepsilon_d \geq 0$. Given the ride-matching mapping mechanism $\mathcal{C}_{Z|Y}$ satisfies $\varepsilon_d/2$-DP. We have PADP-RoD satisfies ε_d-DP for any dynamic pricing mapping mechanism $\mathcal{B}_{Y|X}$.

Proof: Because $\mathcal{A}_{Z|X}(z \mid x) = \sum_{y\in\mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) \cdot \mathcal{B}_{Y|X}(y \mid x)$, we have

$$\frac{\mathcal{A}_{Z|X}(z \mid x)}{\mathcal{A}_{Z|X}(z \mid x')} = \frac{\sum_{y\in\mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) \cdot \mathcal{B}_{Y|X}(y \mid x)}{\sum_{y\in\mathcal{L}} \mathcal{C}_{Z|Y}(z \mid y) \cdot \mathcal{B}_{Y|X}(y \mid x')}, \tag{43}$$

where locations x and x' are neighbors. Since the mechanism $\mathcal{C}_{Z|Y}$ satisfies $\varepsilon_d/2$-DP, we have

$$\mathcal{C}_{Z|Y}(z \mid y) \leq e^{\varepsilon_d/2} \cdot \mathcal{C}_{Z|Y}(z \mid y'), \forall y, y' : y \sim y', z \in \mathcal{L}, \tag{44}$$

$$e^{-\varepsilon_d/2} \cdot \mathcal{C}_{Z|Y}(z \mid y') \leq \mathcal{C}_{Z|Y}(z \mid y), \forall y, y' : y \sim y', z \in \mathcal{L}. \tag{45}$$

According to (44) and (45), we have

$$\frac{\mathcal{A}_{Z|X}(z \mid x)}{\mathcal{A}_{Z|X}(z \mid x')} \leq \frac{\sum_{y\in\mathcal{L}} e^{\varepsilon_d/2} \cdot \mathcal{C}_{Z|Y}(z \mid y') \cdot \mathcal{B}_{Y|X}(y \mid x)}{\sum_{y\in\mathcal{L}} e^{-\varepsilon_d/2} \cdot \mathcal{C}_{Z|Y}(z \mid y') \cdot \mathcal{B}_{Y|X}(y \mid x')}, \tag{46}$$

$$= e^{\varepsilon_d} \times \frac{\sum_{y\in\mathcal{L}} \mathcal{B}_{Y|X}(y \mid x)}{\sum_{y\in\mathcal{L}} \mathcal{B}_{Y|X}(y \mid x')}, \tag{47}$$

$$\overset{(a)}{=} e^{\varepsilon_d}, \tag{48}$$

where (a) follows the fact that $\sum_{y\in\mathcal{L}} \mathcal{B}_{Y|X}(y \mid x) = 1 = \mathcal{B}_{Y|X}(y \mid x')$. In summary, the local mapping mechanism $\mathcal{A}_{Z|X}$ satisfies ε_d-DP. □

Theorem 3: The proposed PADP-RoD satisfies both ε_d-DP and ε_d-identifiability.

Proof: The DP constraints of sub-problems $(\mathcal{O}_d)$ and $(\mathcal{O}_r)$, i.e., constraints (15b) and (25b), guarantee that the dynamic pricing mapping mechanism $\mathcal{B}_{Y|X}$ and the ride-matching mapping mechanism $\mathcal{C}_{Z|Y}$ satisfies ε_d-DP and $\varepsilon_d/2$-DP, respectively. Thus, the local mapping mechanism $\mathcal{A}_{Z|X}$ satisfies ε_d-DP according to the Theorems 1 and 2. Besides, it is known from [33, Theorem 1] that if the mechanism $\mathcal{A}_{Z|X}$ satisfies ε_d-DP then it also satisfies ε_d-identifiability. In summary, the mechanism $\mathcal{A}_{Z|X}$ satisfies both ε_d-DP and ε_d-identifiability. □

B. Computational Complexity and Limitations

The computational complexity of PADP-RoD is composed of two parts: the discretization of spatio-temporal region (i.e., obtain the adaptive supply and demand aware grid) and the solution of problem $(\mathcal{O})$ (i.e., solve the dynamic pricing and ride-matching sub-problems). The complexity of obtaining the grid $\mathbb{G}$ is $O(|T| \times |\mathcal{L}|)$, where $|T| = (ref_{per})^{ref-1} \cdot |T_o|$ and T_o is an initialized temporal grid. Particularly, the complexity of CVT strategy (i.e., Algorithm 1) is $O(|\mathcal{L}|)$ and the complexity of dichotomy strategy (i.e., lines 2 to 6in Algorithm 2) is formulated as

$$\begin{aligned} &O\left(|T_o| \cdot \log|T_o| + ref_{per} \cdot |T_o| \cdot \log(ref_{per} \cdot |T_o|)\right. \\ &\left. + \cdots + (ref_{per})^{ref-1} \cdot |T_o| \cdot \log((ref_{per})^{ref-1} \cdot |T_o|)\right) \\ &\approx O\left(|T_o| \times (\log|T_o| + 1)\right). \end{aligned} \tag{49}$$

Because the spatial region is re-discretize at time $\forall t_i \in T$ by CVT strategy (lines 7 to 11 in Algorithm 2), the complexity of obtaining the grid $\mathbb{G}$ is $O(|T_o| \times (\log|T_o| + 1) + |T| \times |\mathcal{L}|) \approx O(|T| \times |\mathcal{L}|)$.

The complexity of solving the sub-problems $(\mathcal{O}_d)$ and $(\mathcal{O}_r)$ are $O(|\mathcal{L}| \times |\mathcal{L}|)$ and $O(S \times |\mathcal{L}|)$, respectively. Particularly, the complexity of solving the sub-problem $(\mathcal{O}_r)$ mainly comes from Algorithm 5 (line 16 in Algorithm 4) because Algorithm 4 tends to converge within a few iterations. Since the complexity of Algorithm 5 is $O(S \times |\mathcal{L}|)$, the complexity of obtaining the mapping mechanism $\mathcal{C}_{Z|Y}$ is $O(S \times |\mathcal{L}|)$. After the above discussion, the complexity of PADP-RoD is $O((|T| + |\mathcal{L}| + S) \times |\mathcal{L}|)$, and can be simplified to $O(|\mathcal{L}| \times |\mathcal{L}|)$ when $|\mathcal{L}| \gg |T| + S$. We can obtain the local mapping mechanism $\mathcal{A}_{Z|X}$ in the pre-computation process, and then efficiently perturb users' real location locally.

We demonstrate that the PADP-RoD can protect users' location data privacy (by Theorems 1, 2, and 3) while providing high-quality dynamic pricing and ride-matching services (by experimental results). However, there are still two limitations that could be addressed in future research. The first limitation is the lack of theory analysis regarding the bounds of dynamic pricing and ride-matching utility. Second, we propose the dynamic pricing utility metric from a local perspective, measuring the local utility loss caused by perturbation. For the second limitation, we will explore how to measure the global utility loss of dynamic pricing in the local scenario, even though this may be challenging.

VI. Performance Evaluation

In this section, we evaluate the utility (i.e., dynamic pricing utility and ride-matching utility) and location privacy of PADP-RoD on the real-world event-log dataset from an RoD service.

A. Experimental Setup

All experiments are conducted using the same dataset and implementation environment. The simulations are implemented in Python and performed on a laptop with an Intel Core i7-8750H 2.20 GHz processor and 16GB RAM.

1) Dataset: The real-world event-log dataset is collected from Shenzhou UCar,[1] one of the major RoD service providers in China. Our dataset contains 14,587,353 entries in the complete 4 months from Aug to Nov 2016in Beijing, and all entries are properly anonymized. Each entry corresponds to a single event and includes the *Event_time*, *Event_boarding_location (longitude and latitude)*, *Event_arriving_location (longitude and latitude)*, *Estimated_fare*, and *Price_multiplier*. We find that the Shenzhou UCar sets the lower and upper bounds for the price multiplier (1.0 and 1.6), and all possible multipliers are 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, and 1.6. Table II shows two sampled entries in our event-log dataset. Besides, we also collected the GPS locations of drivers during the same period from Shenzhou UCar. Each record includes the *Record_time*, *Driver_ID*, *Lon*, and *Lat*. Table III shows two sampled GPS records in the GPS dataset.

2) Baselines: We evaluate PADP-RoD by comparing it with three literature works, i.e., *Dif-Dis [16], MAL-Geo-I [37], and Laplace [29]*. Next, we briefly introduce these works.

- *Dif-Dis [16]:* Dif-Dis provides high-quality ride-matching service for drivers while protecting their location privacy with geo-indistinguishability [29] and distortion privacy. Particularly, Dif-Dis uses the average travel distance, which is calculated based on the rider's real location, the driver's perturbed location, and the mapping mechanism, to estimate the real geo-distance. Then, Dif-Dis is formulated as the single-objective optimization problem that minimizes the average travel distance given the geo-indistinguishability and distortion privacy constraints.
- *MAL-Geo-I [37]:* MAL-Geo-I aims to minimize the average loss while providing location privacy protection that satisfies geo-indistinguishability [29].
- *Laplace [29]:* Laplace adds quantitative Laplace noise to the real location so that it critically satisfies geo-indistinguishability [29]. The obfuscation probabilities are

[1][Online]. Available: http://www.10101111.com

TABLE II
SAMPLE ENTRIES IN OUR EVENT-LOG DATASET

Event_time	Boarding_lon	Boarding_lat	Arriving_lon	Arriving_lat	Estimated_fare	Price_multiplier
01/11/2016 09:09	116.4658203	39.8971405	116.4675064	39.9571304	44	1.0
30/11/2016 14:58	116.1534729	39.8046532	116.3790894	39.8648796	344	1.6

TABLE III
SAMPLE RECORDS IN OUR GPS DATASET

Record_time	Driver_ID	Lon	Lat
01/11/2016 00:04	NV7H87	116.733986	40.20458
01/11/2016 23:59	QX2F69	116.48728	39.9444

formulated as

$$\mathcal{A}_{Z|X}(z \mid x) \propto e^{-\varepsilon_d \cdot \frac{d(x,z)}{D(\mathcal{L})}}, \tag{50}$$

where $D(\mathcal{L})$ represents the maximum geo-distance between locations in $\mathcal{L}$.

3) Metrics: We evaluate PADP-RoD from the following four aspects: the adaptive supply and demand aware grid, the dynamic pricing utility, the ride-matching utility, and the location privacy.

Adaptive Supply and Demand Aware Grid: Average Price Multiplier Error (APME) is utilized to measure the quality of grids, and can be formulated as

$$\text{APME} = \frac{\sqrt{\sum_{l_i,t_j \in D_{real}} \left(m(l_i,t_j) - \mathbb{E}_m(\mathbb{G}(l_i,t_j))\right)^2}}{|D_{real}|}, \tag{51}$$

where $m(l_i,t_j)$ represents the price multiplier at (l_i,t_j), $\mathbb{G}(l_i,t_j)$ returns the cell (e.g., cell C_n^k) where (l_i,t_j) is located, and $\mathbb{E}_m(*)$ is the average price multiplier on the cell C_n^k. Obviously, the lower APME, the better the grid quality.

Dynamic Pricing Utility: The price multiplier is determined by the current supply-demand relationship. Thus, we adopt Average Supply-Demand Relationship Error (ASDRE) to measure the dynamic pricing utility, and its detailed definition is given below.

Definition 12 (Average Supply-Demand Relationship Error (ASDRE)): Let $S(x,t)/D(x,t)$ (or $S^*(x,t)/D^*(x,t)$) represents the number of real (or perturbed) drivers'/riders' locations in a circular area centered at the location x during period t. We define the supply-demand relationship error as the difference between the real supply-demand relationship (i.e., $S(x,t) - D(x,t)$) and the perturbed supply-demand relationship (i.e., $S^*(x,t) - D^*(x,t)$). The average supply-demand relationship error is formulated as

$$\text{ASDRE} = \frac{1}{|T| \cdot |\mathcal{L}|} \times \sum_{t \in T} \sum_{x \in \mathcal{L}} \Big| (S(x,t) - D(x,t)) - (S^*(x,t) - D^*(x,t)) \Big|. \tag{52}$$

Definition 12 presents the lower ASDRE, the better the obtained dynamic pricing utility.

Ride-Matching Utility: We utilize RMUM (detailed in Definition 10) to measure the ride-matching utility, and the better ride-matching utility when the lower RMUM.

Location Privacy: As a standard location privacy metric, Average Bayesian Inference Error (ABIE) is widely used to measure the effectiveness of defending against Bayesian inference attacks and can be defined as follows.

Definition 13 (Average Bayesian Inference Error (ABIE)): We define the Bayesian inference error as the geo-distance between the inferred location $\widehat{x}$ and real location x of the observed location z (detailed in definition 1). Then, the average Bayesian inference error can be formulated as

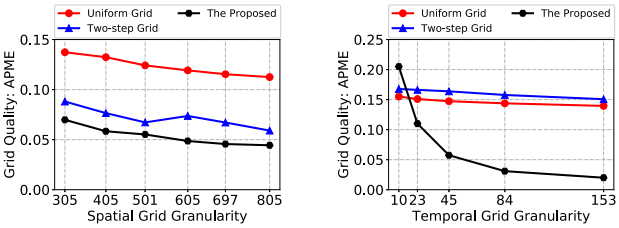
$$\text{ABIE} = \frac{1}{D_{real}} \sum_{z \in D_{real}} d(\widehat{x}, x). \tag{53}$$

Obviously, the better privacy guarantee when the higher ABIE.

4) Parameter Settings: As presented in Section IV, we need to consider the impact of several important parameters: the differential privacy budget ε_d, the number of generators g_n, and the refinement number ref. To evaluate PADP-RoD, we set differential privacy budget $\varepsilon_d = \{0.1, 0.3, 0.5, 0.7, 0.9\}$, the number of generators $g_n = \{50, 100, 150, 200, 250, 300\}$, and the refinement number $ref = \{6, 12, 18, 24, 30\}$. Besides, there are some empirical parameters in PADP-RoD that need to be set. In Algorithm 1, we set the threshold for convergence $\eta = 0.01$ and the percentage of cells that need to be refined $ref_{per} = 0.1$. In Algorithm 3, we set the threshold for convergence $\theta_1 = 0.01$. In Algorithm 4, following [34], we set the threshold for convergence $\theta_2 = 0.01$, the penalty factor $\delta = 10$, the penalty factor growth coefficient $\rho = 10$, and the maximum dynamic pricing utility loss that allows damage $\Phi = 0.1 \cdot \arg\min Pri(\mathcal{B}_{Y|X})$. In Algorithm 5, following [36], we set the swarm size $S = 100$, the maximum number of iterations $MNI = 1000$, and the cognitive and social acceleration coefficients $d_1 = 2, d_2 = 2$. Additionally, we divide a target area Ω into 114×90 cells (each cell size is 500 m $\times$ 500 m), and then the locations set $\mathcal{L}$ is formed by the centers of all cells.

B. Experimental Results

1) Quality of the Adaptive Supply and Demand Aware Grid: We evaluate the quality of the proposed adaptive supply and demand aware grid through comparison with the two-step grid [17] and the uniform grid [31]. The uniform grid technique divides the spatial region Ω and the temporal region T into uniform-size cells. The two-step grid technique divides Ω and T into uniform-size cells and then subdivides each cell into varying sizes to reflect the price multiplier rate-of-change. The evaluation results are shown in Fig. 6, in our dataset, the proposed adaptive supply



(a) Impact of g_n on grid quality (b) Impact of ref on grid quality

Fig. 6. The quality of different grids in a real-word event-log dataset varying with the number of cells. Spatial grid granularity (x axis) represents the number of spatial cells, and $g_n = \{50, 100, 150, 200, 250, 300\}$ means spatial grid granularity: $\{305, 405, 501, 605, 697, 805\}$. Similarly, temporal grid granularity (x axis) represents the number of temporal cells, and $ref = \{6, 12, 18, 24, 30\}$ means temporal grid granularity: $\{10, 23, 45, 84, 153\}$. (a) Average price multiplier error varying with g_n. (b) Average price multiplier error varying with ref.

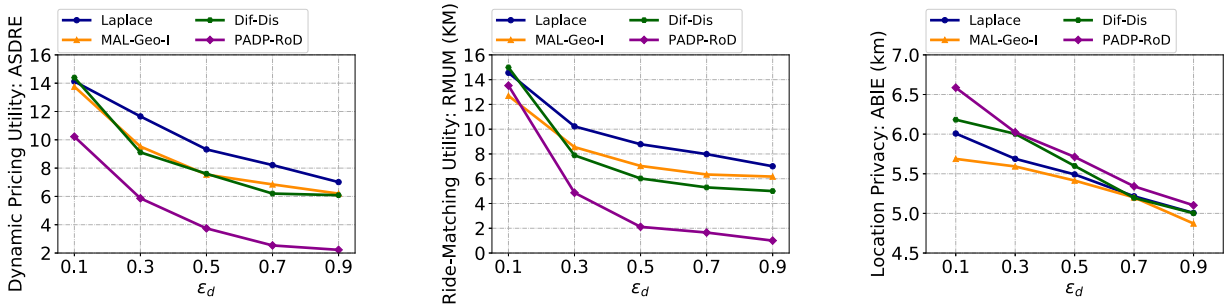
and demand aware grid outperforms the two-step grid and the uniform grid, our Average Price Multiplier Error (APME) is significantly less than that of the two-step grid and the uniform grid. For example, in the spatial dimension, the APME value of ours, the uniform grid, and the two step grid are 0.07, 0.14, and 0.09, respectively, when the number of generators $g_n = 50$. As a result, we can obtain lower APME with fewer cells, and the superiority of our proposed grid is attributed to its ability to adaptively adjust the size of cells according to the price multiplier elasticity.

2) Dynamic Pricing Utility Guarantee: Fig. 7(a) shows the dynamic pricing utility provided by PADP-RoD and all baselines under the real-world event-log dataset from Shenzhou UCar. It could be observed that PADP-RoD outperforms Dif-Dis, MAL-Geo-I and Laplace with the Average Supply-Demand Relationship Error (ASDRE) significantly less than that of Dif-Dis, MAL-Geo-I, and Laplace. For instance, when the privacy budget $\varepsilon_d = 0.9$, the ASDRE value of PADP-RoD, Laplace, Dif-Dis, and MAL-Geo-I are 2.22, 7.01, 6.20, and 6.09, respectively. The superiority of PADP-RoD is attributed to the fact that the Laplace, Dif-Dis, and MAL-Geo-I overlook the damage caused by noise perturbed to the quality of dynamic pricing service. Furthermore, as shown in Figs. 8(a) and (b) (red solid line), we investigate the impact of the number of generators g_n and the refinement number ref on the dynamic pricing utility. We observe that ASDRE decreases sharply when g_n increases from 50 to 200, and saturates afterward. Similarly, increasing ref from 6 to 18 leads to a sharp decrease in ASDRE. This occurs because the increased number of cells can better capture the dynamic nature of price multiplier, but as the number of cells is sufficiently large, the improvement in capturing these natures begins to saturate (detailed in Fig. 6). Finally, Fig. 8(c) (red solid line versus red dashed line) also shows that PADP-RoD and PADP-RoD-one (which consist of dynamic pricing mapping mechanism $\mathcal{B}_{Y|X}$ only) are close in terms of ASDRE, verifying that the ride-matching mapping mechanism $\mathcal{C}_{Z|Y}$ does not excessively harm the quality of dynamic pricing service.

3) Ride-Matching Utility Guarantee: Fig. 7(b) further shows the trade-off between ride-matching utility and location privacy of PADP-RoD and all compared methods over our dataset. It could be observed that PADP-RoD significantly outperforms Dif-Dis, MAL-Geo-I, and Laplace in terms of ride-matching utility. For example, when the privacy budget $\varepsilon_d = 0.9$, the RMUM value of PADP-RoD, Laplace, Dif-Dis, and MAL-Geo-I are 1.00, 7.00, 6.17, and 5.00, respectively. The superiority of PADP-RoD is attributed to MAL-Geo-I and Laplace ignoring the quality losses of ride-matching service caused by noise perturbation, and while MAL-Geo-I estimates the geo-distance between real locations using the expected travel distance, it does not attempt to reduce the errors between these distances. Moreover, we also investigate the impact of the number of generators g_n and the refinement number ref on the ride-matching utility (measured by the proposed ride-matching utility metric), which is shown in Fig. 8(a) and (b) (blue solid line). It shows that the Ride-Matching Utility Metric (RMUM) increases slowly when g_n increases from 50 to 200, and increases sharply afterward. The ride-matching utility does not vary with ref increases, because ref does not affect the ride-matching utility (detailed in Section III). These results demonstrate that an excessive number of generators g_n (i.e., when the size of the adaptive grid is too small) leads to poorer ride-matching utility. This occurs due to the challenge of finding suitable perturbed locations in excessively small cells, which hampers efforts to reduce the service quality losses caused by noise in ride-matching. Lastly, Fig. 8(c) (blue solid line versus blue dashed line) also shows that PADP-RoD outperforms PADP-RoD-one (i.e., the local mapping mechanism that contains only dynamic pricing mapping mechanism $\mathcal{B}_{Y|X}$) in terms of ride-matching utility with the RMUM significantly less than that of PADP-RoD-one, verifying that the dynamic pricing mapping mechanism $\mathcal{B}_{Y|X}$ cannot provide high-quality ride-matching service and the ride-matching mapping mechanism $\mathcal{C}_{Z|Y}$ is necessary.

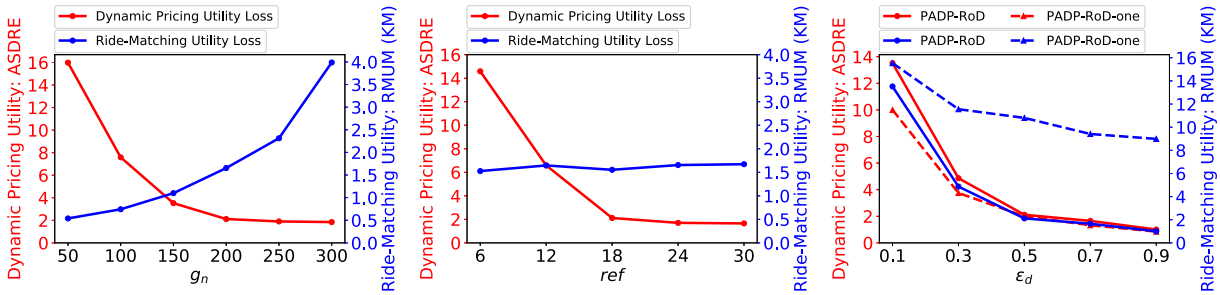
4) Effectiveness of Privacy Protection: The location privacy protection provided by PADP-RoD is evaluated under our dataset. Fig. 7(c) shows that the effectiveness of PADP-RoD and all baselines defend against Bayesian inference attack. It could be observed that PADP-RoD, Dif-Dis, MAL-Geo-I, and Laplace provide comparable privacy protection to resist Bayesian inference attack. For instance, when the privacy budget $\varepsilon_d = 0.9$, the ABIE value of PADP-RoD, Laplace, Dif-Dis, and MAL-Geo-I are 5.10, 5.00, 4.87, and 5.01, respectively. This is because PADP-RoD and all baselines satisfy ε_d-differential privacy or ε_d-Geo-indistinguishability (a generalized version of differential privacy). Furthermore, we also investigate the impact of the differential privacy budget ε_d, the number of generators g_n, and the refinement number ref on the location privacy, as shown in Fig. 9 (the magenta line: PADP-RoD). It shows that the Average Bayesian Inference Error (ABIE) decreases as privacy budget ε_d increases, i.e., the effectiveness of location privacy protection decreases as privacy budget ε_d increases. The number of generators g_n and the refinement number ref only affect the dynamic pricing and ride-matching utilities (detailed in Section II-C). As a result, ABIE is stable with the variation of g_n and ref.

Finally, we propose PADP-RoD-AOne and PADP-RoD-ATwo, aiming to validate Theorem 1, Theorem 2, and Theorem 3. Specifically, PADP-RoD-AOne consists of an optimal



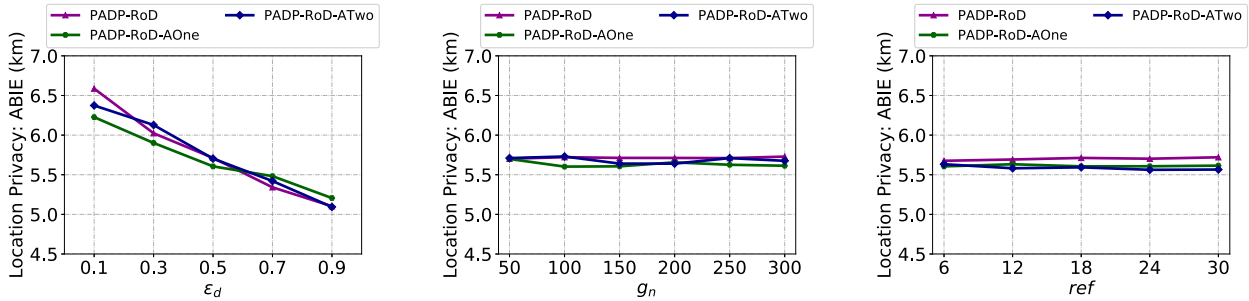
(a) Impact of ε_d on dynamic pricing utility (b) Impact of ε_d on ride-matching utility (c) Impact of ε_d on location privacy

Fig. 7. (a) Dynamic pricing utility loss of different location-privacy preserving mechanisms. (b) Ride-Matching utility loss of different location-privacy preserving mechanisms. (c) Effectiveness of different location privacy-preserving mechanisms defends against the Bayesian inference attack.



(a) Impact of g_n on utility loss (b) Impact of ref on utility loss (c) Impact of ε_d on utility loss

Fig. 8. Utility loss (the red line represents the dynamic pricing utility loss and the blue line represents the ride-matching utility loss) in a real-word event-log dataset varying with the number of generators g_n, the refinement number ref, and the differential privacy budget ε_d. The dynamic pricing utility loss is measured by the Average Supply-Demand Relationship Error (ASDRE) and the ride-matching utility loss is measured by the Ride-Matching Utility Metric (RMUM). (a) Utility loss varies with g_n. (b) Utility loss varies with ref. (c) Utility loss varies with ε_d, where the comparison of PADP-RoD and PADP-RoD-one (consists only of dynamic pricing mapping mechanism $\mathcal{B}_{Y|X}$) is used to evaluate whether the mapping mechanism $\mathcal{B}_{Y|X}$ can preserve the ride-matching utility and whether the mapping mechanism $\mathcal{C}_{Z|Y}$ can damage the dynamic pricing utility.



(a) Impact of ε_d on location privacy (b) Impact of g_n on location privacy (c) Impact of ref on location privacy

Fig. 9. Average Bayesian Inference Error (ABIE) in a real-word event-log dataset varying with the differential privacy budget ε_d, the number of generators g_n, and the refinement number ref. PADP-RoD-AOne consists of an optimal stage-one privacy mapping mechanism $\mathcal{B}_{Y|X}$ and an arbitrary stage-two privacy mapping mechanism $\mathcal{C}_{Z|Y}$. Similarly, PADP-RoD-ATwo consists of an arbitrary $\mathcal{B}_{Y|X}$ and an optimal $\mathcal{C}_{Z|Y}$. Additionally, our propose PADP-RoD consists of the optimal $\mathcal{B}_{Y|X}$ and $\mathcal{C}_{Z|Y}$. (a) ABIE varies with ε_d. (b) ABIE varies with g_n. (c) ABIE varies with ref.

dynamic pricing mapping mechanism $\mathcal{B}_{Y|X}$ and an arbitrary ride-matching mapping mechanism $\mathcal{C}_{Z|Y}$. PADP-RoD-ATwo consists of an arbitrary dynamic pricing mapping mechanism $\mathcal{B}_{Y|X}$ and an optimal ride-matching mapping mechanism $\mathcal{C}_{Z|Y}$. The experimental results are shown in Fig. 9. It shows that PADP-RoD-AOne, PADP-RoD-ATwo, and PADP-RoD (i.e., it consists of the optimal $\mathcal{B}_{Y|X}$ and $\mathcal{C}_{Z|Y}$) provide comparable privacy protection to resist Bayesian inference attack, which is consistent with our privacy analysis results (i.e., Theorem 1, Theorem 2, and Theorem 3).

VII. Related Work

Many privacy-preserving techniques have been proposed to protect users' location privacy, and we review the RoD services with dynamic pricing and the privacy-preserving ride-matching service. These works focus on three dimensions: *Dynamic Pricing Utility* (providing high-quality dynamic pricing service), *Data Privacy* (protecting users' location data privacy), and *Ride-Matching Utility* (providing high-quality ride-matching service). As shown in Table IV, previous works mainly fall

TABLE IV
RELATED WORKS ON RIDE-ON-DEMAND SERVICES

Works	**Dynamic Pricing Utility**: Providing High-Quality Dynamic Pricing Service	**Data Privacy**: Protecting Users' Location Data Privacy	**Ride-Matching Utility**: Providing High-Quality Ride-Matching Service
[2]	✓ Dynamic pricing service is crucial.	×	×
[14]	×	✓ Protecting location data privacy while providing high-quality ride-matching service.	✓
Ours	✓ Protecting location data privacy while providing high-quality dynamic pricing and ride-matching services.	✓	✓

into two topics. First, the authors design dynamic pricing strategies from the service providers' perspective or propose dynamic pricing prediction approaches from the users' viewpoint, focusing on providing high-quality dynamic pricing service (which emphasizes that dynamic pricing service is crucial). Second, the authors focus on the trade-off between data privacy and ride-matching utility, proposing various privacy-preserving mechanisms that protect location data privacy while providing high-quality ride-matching service (which ignores the damage caused by noise perturbation to dynamic pricing service). Different from these works, we propose a pricing-aware differentially private framework, protecting users' location data privacy while providing them with high-quality dynamic pricing and ride-matching services.

A. RoD Services With Dynamic Pricing

RoD services as on-demand ride-hailing services can adaptively adjust supply and demand through dynamic pricing. In recent years, a lot of works have been done around dynamic pricing [1], [2], [3], [4], [5], [6], [7]. For example, Guo et al. [2] studied the process of searching for the best price before getting on a car, and the searching process reflects the riders' reaction to dynamic prices. Guo et al. [1] studied the effect of traditional counterparts (e.g., metro, bus, and taxi) on the dynamic prices, and proposed a dynamic price prediction approach based on multi-source urban data. Guo et al. [3] studied how drivers seek riders to earn more under the dynamic pricing mechanism, and proposed the RoD-Revenue to predict drivers' revenue based on multi-source urban data. Guo et al. [4] studied how riders maximize revenue under the dynamic pricing mechanism, and proposed the route recommendation approach based on multi-source urban data. Additionally, some works have investigated better dynamic pricing strategies from the view of economics [5], [6], [7]. For instance, Huang et al. [6] proposed the dynamic pricing scheme based on deep reinforcement learning to better balance the demands and supplies. Shao et al. [7] devised a novel dynamic (or fair) pricing policy under asymmetrical demand patterns. As a result, dynamic pricing plays a very important role in RoD services, and is designed based on supply and demand. *To protect users' location privacy, perturbation-based privacy-preserving mechanisms are widely used in RoD services. However, perturbation leads to severe dynamic pricing utility loss because it disrupts the supply-demand relationship within the RoD service area.*

B. Privacy-Preserving Ride-Matching Service

Although ride-matching is an essential service in RoD services, it is seriously damaged by obfuscation. Thus, a lot of works [14], [15], [16], [38], [39], [40] have been proposed to protect users' location privacy while providing them with high-quality ride-matching service. For example, Wang et al. [14] used the average travel distance of selected drivers (calculated based on riders' real location and drivers' perturbed location) instead of real geo-distance in the ride-matching service. Then, finding a driver closest to the rider (i.e., ride-matching service) is formalized as a mixed-integer non-linear programming problem that minimizes selected drivers' travel distance to the rider's location given the constraint of differential privacy. Wang et al. [15] proposed a novel location obfuscation mechanism that is formulated as a single-object minimization problem and aims to minimize the utility loss caused by perturbation under the constraints of ε-differential privacy and δ-distortion privacy. Following these works, Wang et al. [16] proposed an optimal task allocation framework with the differential-and-distortion.

However, estimating the real geo-distance based on known information such as the observed locations and the privacy-preserving mechanism is inaccurate, which leads to lower-quality ride-matching service. To improve the quality of ride-matching service, some works [38], [39], [40] have designed various privacy-preserving mechanisms that do not change the ranking of real locations' relative distance. For example, To et al. [40] proposed a framework for assigning tasks (or riders) to workers (or drivers) in an online manner without compromising the location privacy of both workers and tasks. Particularly, they designed a novel privacy mechanism based on hierarchically well-separated trees to preserve the ranking of real locations' relative distance. Following this work, Wang et al. [38] propose a planar Laplace distribution-based privacy mechanism that does not change the ranking of these locations' relative distances, and then design a threshold-based online task assignment mechanism that could deal with the one-worker-many-tasks assignment. As shown in Fig. 1, however, it is difficult to achieve high-quality ride-matching service by just keeping the ranking unchanged. *Different from existing works, PADP-RoD estimates the real geo-distance by the proposed expected travel distance, and then minimizes the estimated error (i.e., the difference between the expected travel distance and the geo-distance) caused by obfuscation under DP constraint. In this way, PADP-RoD can provide users with high-quality service without leaking the users' location privacy.*

VIII. Conclusion

In this paper, we have proposed a pricing-aware differentially private framework for ride-on-demand services named PADP-RoD. PADP-RoD can provide users with high-quality dynamic pricing and ride-matching services while protecting their location privacy. Moreover, the proposed dynamic pricing and ride-matching utility metrics are generic, and can accurately measure quality loss caused by perturbation in the sporadic location privacy scenarios. Importantly, PADP-RoD can be adapted to other location-based applications such as mobile crowdsensing to improve service quality while protecting users' location privacy. For the future work, we will investigate a non-sporadic location privacy protection mechanism, which can protect users' location traces privacy while providing them with high-quality location-based services.

References

[1] S. Guo et al., "A simple but quantifiable approach to dynamic price prediction in ride-on-demand services leveraging multi-source urban data," in *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 3, Sep. 2018, Art. no. 112.

[2] S. Guo, C. Chen, Y. Liu, K. Xu, and D. M. Chiu, "Modelling passengers' reaction to dynamic prices in ride-on-demand services: A search for the best fare," in *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 4, Jan. 2018, Art. no. 136.

[3] S. Guo et al., "ROD-revenue: Seeking strategies analysis and revenue prediction in ride-on-demand service using multi-source urban data," *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2202–2220, Sep. 2020.

[4] S. Guo et al., "A force-directed approach to seeking route recommendation in ride-on-demand service using multi-source urban data," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 1909–1926, Jun. 2022.

[5] M. K. Chen, "Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform," in *Proc. ACM Conf. Econ. Comput.*, 2016, pp. 1–15.

[6] J. Huang et al., "Deep reinforcement learning-based trajectory pricing on ride-hailing platforms," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 3, Mar. 2022, Art. no. 41.

[7] S. Shao, A. Mittal, R. Twumasi-Boakye, and A. Gupta, "Fair pricing of ridehailing services with asymmetric demand and travel time," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 2, pp. 670–681, Jun. 2022.

[8] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services: A comprehensive survey," *ACM Comput. Surv.*, vol. 54, no. 1, Jan. 2021, Art. no. 4.

[9] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Sci. Rep.*, vol. 3, no. 1, pp. 2045–2322, 2013.

[10] Z. Zheng, Z. Li, H. Jiang, L. Y. Zhang, and D. Tu, "Semantic-aware privacy-preserving online location trajectory data sharing," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 2256–2271, 2022.

[11] Z. Zheng, Z. Li, J. Li, H. Jiang, T. Li, and B. Guo, "Utility-aware and privacy-preserving trajectory synthesis model that resists social relationship privacy attacks," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 3, May 2022, Art. no. 44.

[12] Y. Luo, X. Jia, S. Fu, and M. Xu, "pRide: Privacy-preserving ride matching over road networks for online ride-hailing service," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 7, pp. 1791–1802, Jul. 2019.

[13] J. Huang, Y. Luo, S. Fu, M. Xu, and B. Hu, "pRide: Privacy-preserving online ride hailing matching system with prediction," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 7413–7425, Aug. 2021.

[14] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, and X. Ma, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 627–636.

[15] L. Wang, D. Zhang, D. Yang, B. Y. Lim, X. Han, and X. Ma, "Sparse mobile crowdsensing with differential and distortion location privacy," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2735–2749, 2020.

[16] L. Wang, D. Yang, X. Han, D. Zhang, and X. Ma, "Mobile crowdsourcing task allocation with differential-and-distortion geo-obfuscation," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 967–981, Mar. 2021.

[17] M. E. Gursoy, L. Liu, S. Truex, L. Yu, and W. Wei, "Utility-aware synthesis of differentially private and attack-resilient location traces," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 196–211.

[18] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *Proc. IEEE Symp. Secur. Privacy*, 2016, pp. 546–563.

[19] Z. Zheng, Z. Li, C. Huang, S. Long, M. Li, and X. Shen, "Data poisoning attacks and defenses to LDP-based privacy-preserving crowdsensing," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 5, pp. 4861–4878, Sep./Oct. 2024.

[20] Z. Li, Z. Zheng, S. Guo, B. Guo, F. Xiao, and K. Ren, "Disguised as privacy: Data poisoning attacks against differentially private crowdsensing systems," *IEEE Trans. Mobile Comput.*, vol. 22, no. 9, pp. 5155–5169, Sep. 2023.

[21] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE 54th Annu. Symp. Found. Comput. Sci.*, 2013, pp. 429–438.

[22] BBC, "Didi shares fall on reports China is planning penalties," Jul. 2021. [Online]. Available: https://www.bbc.com/news/business-57938212

[23] Y. Tong, J. She, B. Ding, L. Chen, T. Wo, and K. Xu, "Online minimum matching in real-time spatial data: Experiments and analysis," in *Proc. VLDB Endowment*, vol. 9, no. 12, pp. 1053–1064, 2016.

[24] P. Xu et al., "A unified approach to online matching with conflict-aware constraints," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 2221–2228.

[25] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: Optimal strategy against localization attacks," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2012, pp. 617–627.

[26] S. Oya, C. Troncoso, and F. Pérez-González, "Rethinking location privacy for unknown mobility behaviors," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2019, pp. 416–431.

[27] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *J. Privacy Confidentiality*, vol. 7, no. 3, pp. 17–51, May 2017.

[28] W. Wang, L. Ying, and J. Zhang, "On the relation between identifiability, differential privacy, and mutual-information privacy," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5018–5029, Sep. 2016.

[29] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 901–914.

[30] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, no. 4, pp. 637–676, 1999.

[31] G. Acs and C. Castelluccia, "A case study: Privacy preserving release of spatio-temporal density in Paris," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 1679–1688.

[32] Q. Zhao, C. Zuo, G. Pellegrino, and Z. Lin, "Geo-locating drivers: A study of sensitive data leakage in ride-hailing services," in *Proc. 26th Annu. Netw. Distrib. System Secur. Symp.*, 2019, pp. 1–15.

[33] M. Sun and W. P. Tay, "On the relationship between inference and data privacy in decentralized IoT networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 852–866, 2020.

[34] C. Price and I. Coope, "An exact penalty function algorithm for semi-infinite programmes," *BIT Numer. Math.*, vol. 30, pp. 723–734, 1990.

[35] G. Gramlich, R. Hettich, and E. W. Sachs, "Local convergence of SQP methods in semi-infinite programming," *SIAM J. Optim.*, vol. 5, no. 3, pp. 641–658, 1995.

[36] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, 2007.

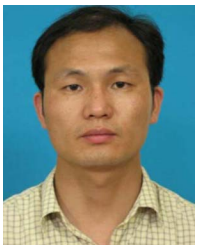
[37] S. Oya, C. Troncoso, and F. Pérez-González, "Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1959–1972.

[38] H. Wang, E. Wang, Y. Yang, J. Wu, and F. Dressler, "Privacy-preserving online task assignment in spatial crowdsourcing: A graph-based approach," in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 570–579.

[39] Q. Tao, Y. Tong, Z. Zhou, Y. Shi, L. Chen, and K. Xu, "Differentially private online task assignment in spatial crowdsourcing: A tree-based approach," in *Proc. IEEE 36th Int. Conf. Data Eng.*, 2020, pp. 517–528.

[40] H. To, C. Shahabi, and L. Xiong, "Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 833–844.

Zhirun Zheng received the BS degree from the School of Mathematics and Statistics, Henan University of Science and Technology, China, in 2017, and the PhD degree from the School of Mathematics and Computational Science, Xiangtan University, China, in 2024. From December 2022 to December 2023, he visited the Broadband Communications Research (BBCR) Lab, Department of Electrical and Computer Engineering, University of Waterloo, Canada. Since 2024, he is a postdoctoral fellow with the Department of Artificial Intelligence, Ajou University, South Korea. His research interests are in the areas of security and privacy in Crowdsensing systems, artificial intelligence, and mobile networks. He serves as TPC members of many conferences, including GLOBECOM and WCSP.



Zhetao Li received the BEng degree in electrical information engineering from Xiangtan University in 2002, the MEng degree in pattern recognition and intelligent systems from Beihang University in 2005, and a PhD degree in computer application technology from Hunan University in 2010. is a professor with the College of Information Science and Technology, Jinan University. From 2013 to 2014, he was a postdoc in wireless networks with Stony Brook University. He is a member of CCF.

Saiqin Long received the PhD degree in computer applications technology from the South China University of Technology, Guangzhou, China, in 2014. She is currently a professor with the College of Information Science and Technology, Jinan University, China. Her research interests include cloud computing, edge computing, parallel and distributed systems, and Internet of things. She has published 20+ refereed papers in these areas, most of which are published in premium conferences and journals, including *IEEE Transactions on Services Computing*, *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Mobile Computing*, etc. She is a member of Chinese Computer Federation (CCF).

Suiming Guo received the PhD degree from the Chinese University of Hong Kong. He is currently an associate professor with the College of Information Science and Technology, Jinan University, Guangzhou, China. His research interests include data mining, urban computing, pervasive computing and smart cities studies.

Chao Chen (Senior Member, IEEE) received the PhD degree from Sorbonne University Pierre and Marie Curie Campus (Paris 6) and Telecom SudParis. He is currently a full professor of computer science with Chongqing University, China. His research interests include pervasive computing, social network analysis, and mobile crowdsensing.

Ke Xu (Fellow, IEEE) received the PhD degree from Tsinghua University. He is currently a full professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include next generation Internet, P2P systems, Internet of Things (IoT), network virtualization and optimization.