Distributed Data-Sharing Consensus in Cooperative Perception of Autonomous Vehicles

*Chenxi Qiu, *Sourabh Yadav, †Anna Squicciarini, *Qing Yang, *Song Fu, ‡Juanjuan Zhao, and ⊥ Chengzhong Xu
*Department of Computer Science and Engineering, University of North Texas, USA
†College of Information Science and Technology, The Pennsylvania State University, USA
‡ Shenzhen Institute of Advanced Technology, P. R. China
⊥ Department of Computer Science, University of Macau, P. R. China

***Abstract*—To enable self-driving without a human driver, an autonomous vehicle needs to perceive its surrounding obstacles using onboard sensors, of which the perception accuracy might be limited by their own sensing range. An effective way to improve vehicles' perception accuracy is to let nearby vehicles exchange their sensor data so that vehicles can detect obstacles beyond their own sensing ranges, called *cooperative perception*. The shared sensor data, however, might disclose the sensitive information of vehicles' passengers, raising privacy and safety concerns (e.g. stalking or sensitive location leakage).**

In this paper, we propose a new data-sharing policy for the cooperative perception of autonomous vehicles, of which the objective is to minimize vehicles' information disclosure without compromising their perception accuracy. Considering vehicles usually have different desires for data-sharing under different traffic environments, our policy provides vehicles autonomy to determine what types of sensor data to share based on their own needs. Moreover, given the dynamics of vehicles' data-sharing decisions, the policy can be adjusted to incentivize vehicles' decisions to converge to the desired decision field, such that a healthy cooperation environment can be maintained in a long term. To achieve such objectives, we analyze the dynamics of vehicles' data-sharing decisions by resorting to the game theory model, and optimize the data-sharing ratio in the policy based on the analytic results. Finally, we carry out an extensive trace-driven simulation to test the performance of the proposed data-sharing policy. The experimental results demonstrate that our policy can help incentivize vehicles' data-sharing decisions to the desired decision fields efficiently and effectively.

***Index Terms*—Autonomous vehicles, cooperative perception, data-sharing**

I. Introduction

Autonomous vehicles are undergoing rapid development, and many autonomous vehicles are already operating on public roads, such as self-driving taxis from Google's Waymo One [1] and self-driving trucks from TuSimple [2]. To enable self-driving without a human driver, an autonomous vehicle needs to perceive its surrounding obstacles via onboard sensors such as cameras, LiDARs (Light Detection and Ranging), and radars. Since perception directly impacts safety-critical driving decisions like route planning and collision avoidance, it is of extra importance for vehicles to ensure high accuracy of perception (for simplicity, we use "autonomous vehicle" and "vehicle" interchangeably in this paper).

While a single vehicle's perception is often limited by the coverage of its own sensors, to improve perception accuracy, an effective strategy is to enable *cooperative perception* among vehicles, such that nearby vehicles can share their collected sensor data with others. In this way, each vehicle can detect objects beyond its own sensing ranges [3], [4]. Vehicles' cooperative perception can be implemented via various well-developed communication technologies for connected vehicles, including Vehicle-to-Vehicle (V2V) [5], Vehicle-to-Infrastructure (V2I) [6], and Vehicle-to-Cloud (V2C) communications [7].

However, when sharing sensor data, vehicles might disclose passengers' or pedestrians' sensitive information (e.g., camera image, location, etc.) to others. For example, the shared sensor data from a vehicle might disclose where and when its passenger travels from, raising privacy and safety concerns (e.g. stalking or sensitive location leakage [8]). According to a recent survey of public opinion about sensor data collection of autonomous vehicles in the U.S., the U.K., and Australia [9], 63.7% of surveyees were either "very concerned" or "moderately concerned" about the information leakage from vehicles. Unfortunately, these privacy concerns so far have not received much attention compared with geo-location privacy concerns of conventional smart vehicles [10]–[15].

To address this research gap, the first objective of this paper is to *design a new data-sharing policy for autonomous vehicles' cooperative perception to minimize vehicles' information disclosure without compromising perception accuracy*. Note that the vehicles' desire for data-sharing varies across different road segments (due to different road features) and different time periods (due to different traffic conditions). For instance, a vehicle might have a higher desire to share data during peak hours, when there are more surrounding vehicles so the vehicle's own sensors are less likely to cover its desired decision field of view [4]. In fact, a uniform data-sharing policy for all vehicles is inefficient, since it does not give vehicles the option to fine-tune the privacy knob based on their own needs. This, on the other hand, presents a barrier for more vehicles to participate in cooperative perception.

As a solution, our data-sharing policy gives vehicles autonomy to control what types of sensor data to share based on their own needs. The more data a vehicle shares with the nearby vehicles, the more data it will collect from others. Moreover, given that vehicles are typically designed to be self-centered and short-sighted, our second objective is to *adjust* *the data-sharing policy dynamically to incent decisions to converge to the desired decision f a healthy cooperation environment can be ma long term.* To achieve this objective, we analyze of vehicles' data-sharing decisions under the p a game-theoretical framework. In the game considered players having to determine what t data to share to meet their perception and priv

Note that analyzing a vehicle's data-sharing non-trivial task. First, vehicles' decisions are dynamic due to the vehicle's mobility, e.g., a frequently change its data-sharing decision w from one road segment to another. Second, dif might have different sensor models, and their pa have different privacy preferences. Harmonizing ual differences can lead to an extremely high co for the policy design/adjustment.

Accordingly, instead of tracking each vehicle's action, we develop an efficient, coarse-grained model to analyze vehicles' *group* behaviors. Specifically, we first partition the whole area into a set of regions such that, in each region, vehicles' desires for data-sharing are similar. This allows us to ignore the decision changes caused by vehicles' mobility within each region. We then apply evolutionary game theory to analyze the proportion of different data-sharing decisions in each region. Here, vehicles with the same decision are considered a "group" (or "species"). This theoretical framework allows us to study how each group evolves when interacting with other groups, based on the group fitness values (fitness values are defined as a combination of safety and privacy). Informed by findings from this game-theoretical formulation, we then design a time-efficient algorithm, called *fast decision shaping (FDS)*, aiming to achieve a healthy cooperative environment for vehicles, i.e., the safety of vehicles can be guaranteed and the privacy of each vehicle can be well-protected.

We carry out a trace-driven simulation to test the performance of our data-sharing policy. The dataset applied in the simulation contains around 700 million GPS positions, timestamps, and the velocity of around 280,000 vehicles (including taxicabs and transit service vehicles) in Shenzhen. The experimental results demonstrate that our FDS can help converge vehicles' data-sharing decisions to desired decision fields efficiently. For theoretical interests, we also derive a lower bound of the convergence time of vehicles' decisions and demonstrate FDS can achieve the optimal closely (e.g., the approximation ratios of FDS are up to 1.15).

Our contributions can be summarized as follows:

1) We design a new data-sharing policy for cooperative perception among vehicles, which allows vehicles to control the types of sensor data to share based on their own needs.
2) We develop a game-theoretical model to analyze the dynamics of vehicles' decision distribution over time, including their decision equilibrium (or evolutionary stable strategies) under different conditions.

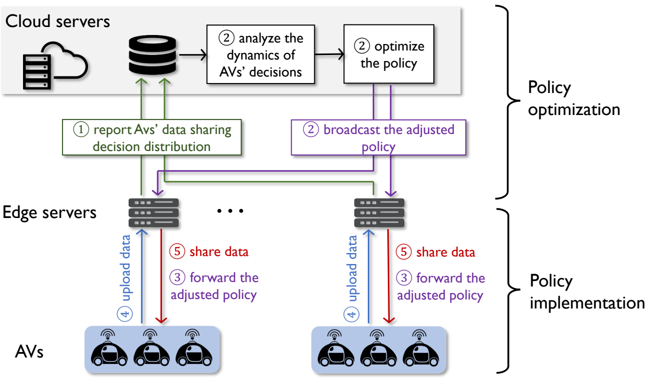

Cloud servers
② analyze the dynamics of AVs' decisions
② optimize the policy
Policy optimization
① report Avs' data sharing decision distribution
② broadcast the adjusted policy
Edge servers
...
④ upload data
⑤ share data
③ forward the adjusted policy
④ upload data
⑤ share data
③ forward the adjusted policy
Policy implementation
AVs

Fig. 1. The policy framework.

3) Informed by our game theoretical results, we then apply a time-efficient policy adjustment strategy to converge vehicles' decisions to the desired decision field.
4) We carry out a comprehensive simulation to demonstrate the effectiveness and efficiency of our policy strategy.

The rest of the paper is organized as follows: the next section introduces the framework of our data-sharing strategy. Section III and Section IV introduce the implementation and the optimization of our data-sharing policy. Section V evaluates the performance of our strategies. Finally, Section VI presents the related work and Section VII makes a conclusion.

II. Framework

Fig. 1 shows the framework of the vehicle cooperative perception system, which includes *vehicles*, *edge servers*, and *cloud servers*. We consider vehicles that use three types of sensor data: *cameras*, *radars*, and *LiDARs* [3], [4]. To improve the perception accuracy, vehicles are allowed to "cooperatively" share sensor data with their nearby vehicles. Here, we assume that vehicles share surrounding data only through edge servers. That is, the vehicles first upload their sensor data to the nearest edge server, of which the location is fixed (sitting together with a road side unit), and the edge server determines how to distribute received sensor data to the vehicles it covers. We do not consider V2V data-sharing in this paper.

As for the threat model, we assume that the server may suffer from a *passive attack wherein attackers can eavesdrop on vehicles' shared sensor data breached by edge servers* [12]. This, unfortunately, may leak passengers' sensitive information, such as locations (by radar and LiDAR data [16]) and identities (by camera [17]). To protect data privacy, a vehicle might choose not to share its own data, or only share part of data that is less likely to disclose sensitive information. For example, when a vehicle might not share its camera data, which might disclose its passengers' identities [17].

In order to achieve a healthy cooperative environment for vehicles, cloud servers optimize the data-sharing policies such that vehicles' overall safety is improved and data privacy is guaranteed. In our framework, we discretize time into rounds

$t = 1, 2, ..., T$, and assume t
sharing do not change within e
the cloud server initializes a dat
the policy to all the edge serv
policy in their respective areas.
sharing process in each round
components, illustrated by Fig.

S1 - *Policy optimization*: Eacl
vehicles' data-sharing dec
cloud servers (Step ①). T
the optimal policy accordi
to each edge server (Step

S2 - *Policy implementation*: Af
from the cloud, edge ser
vehicles they cover (Step
determine what types of sensor data to share with their edge servers (Step ④). According to the policy and the amount of sensor data collected from each vehicle, the edge servers determine the types of sensor data to share with their vehicles (Step ⑤). The data exchange in steps ④ and ⑤ is repeated multiple times before the next updated policy arrives.

Next, we introduce the details of policy implementation (Section III) and policy optimization (Section IV). Table I lists the main notations used throughout this paper.

TABLE I
MAIN NOTATIONS AND THEIR DESCRIPTIONS

Symbol	Description
Ω	The universal data set. Ω_{cam}, Ω_{lid}, and Ω_{rad} represent the universal set of camera data, LiDAR data, and radar data.
h_a	Utility of vehicle a.
c_a	Privacy cost of vehicle a.
r_i	Subregion i.
$\mathcal{G}^t$	Auxiliary graph $\mathcal{G}^t = (\mathcal{R}, \mathcal{E}^t)$ at time t.
$\mathcal{R}$	$\mathcal{R} = \{r_1, ..., r_M\}$, which is also the node set in $\mathcal{G}^t$.
$\mathcal{E}^t$	Edge set in $\mathcal{G}^t$, where each edge $e_{i,j} \in \mathcal{E}^t$ represents the r_i and r_j are neighbors in $\mathcal{G}^t$.
$p_{i,k}^t$	Number of vehicles selecting data-sharing decision k in r_i at time t. $\mathbf{p}_i^t = \left[p_{i,1}^t, ..., p_{i,K}^t\right]$.
x_i	Data sharing ratio in region r_i;

III. Data-Sharing Policy and Its Implementation

In this section, we introduce the details of the data-sharing policy among vehicles and the policy implementation. We assume that each vehicle only uploads its sensor data to the nearest edge server. Accordingly, given the locations of the edge servers, the whole area is partitioned into a set of Voronoi cells [18]. Each cell has one edge server, which is the closest edge server to all the locations within this cell. The data-sharing among vehicles happens independently in each cell. Without loss of generality, in what follows, we focus on the policy implementation in one cell, in which the set of vehicles are denoted by $\mathcal{A}$. The data-sharing in all the other cells follows the same process.

Data-sharing policy. Vehicles might choose different types of sensor data to share given their surrounding environments.

Fig. 2.
of the p

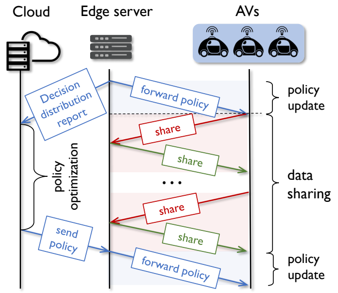


Fig. 3. Illustration of policy implementation.

For example, when the primary goal of a vehicle is to detect its distance from the surrounding objects instead of checking traffic signs, it tends to share/collect the LiDAR/radar data rather than the camera data [19]. We let Ω_{cam}, Ω_{lid}, and Ω_{rad} denote the universal set of camera, LiDAR, and radar data, respectively, and let $\Omega = \{\Omega_{\text{cam}}, \Omega_{\text{lid}}, \Omega_{\text{rad}}\}$. We discretize the data-sharing decision space of vehicles into K decisions, where each decision k specifies the types of data $P^k \subseteq \Omega$ ($k = 1, ..., K$) to share. If a vehicle a selects decision k_a, given its collected data S_a, its shared data should be $S_a^{k_a} = S_a \cap P^{k_a}$. For example, for each type of sensor data (camera, LiDAR, or radar data), if each vehicle has two choices, share or not share, then there are 8 different data-sharing decisions:

$$\begin{aligned}
&P^1 = \{\Omega_{\text{camera}}, \Omega_{\text{lidar}}, \Omega_{\text{radar}}\}, P^2 = \{\Omega_{\text{camera}}, \Omega_{\text{lidar}}\},\\
&P^3 = \{\Omega_{\text{camera}}, \Omega_{\text{radar}}\}, P^4 = \{\Omega_{\text{lidar}}, \Omega_{\text{radar}}\},\\
&P^5 = \{\Omega_{\text{camera}}\}, P^6 = \{\Omega_{\text{lidar}}\}, P^7 = \{\Omega_{\text{radar}}\}, P^8 = \{\}.
\end{aligned}$$

In general, we let $P^1 = \Omega$ (i.e., at decision 1, all the collected data is shared) and $P^K = \phi$ (i.e., at decision K, no data is shared). If $P^l \subsetneq P^k$, then data-sharing decision l is called a **successor** of the decision k, and the decision k is called a **predecessor** of the decision l, denoted by $k \prec l$. We use $k \preceq l$ to denote $k \prec l$ *or* $k = l$.

We apply a lattice-based data-sharing policy [20], defined as follows:

> **Lattice-based data-sharing policy**: Given any two vehicles $a, b \in \mathcal{A}$ sharing their collected data with decision k_a and k_b, respectively. If $k_a \preceq k_b$, vehicle a has probability x to access the shared data from b; otherwise, vehicle a cannot access b's data. Here $x \in [0, 1]$ is called *sharing ratio*, which is controlled by the edge server and cannot be changed within each round.

According to the lattice-based policy, the relationship between the 8 decisions of data-sharing can be described as a directed acyclic graph shown in Fig. 2. In the figure, each edge points from a decision to its successor.

Policy implementation. As Fig. 3 shows, policy implementation is composed of two steps in each round:

1) Policy update: At the beginning, the edge server forwards the policy received from the cloud to all its vehicles.
2) Data-sharing: According to the updated policy, each vehicle determines the amount of sensor data to upload to the edge server. After collecting the sensor data from all the vehicles, the edge server distributes the sensor data to each vehicle according to the data uploaded by the vehicle and the policy. This process is carried out multiple times per round before a new policy is received.

Vehicle's decision analysis. For each vehicle $a \in \mathcal{A}$, we let $D_a \subseteq \Omega$ and $S_a \subseteq \Omega$ denote its *desired sensor data set* and *collected sensor data set*. We quantify the *utility* and the *privacy cost* of vehicle a by the real numbers h_a and c_a, respectively, where both h_a and c_a are normalized to the range of $[0, 1]$. In particular, higher h_a (resp. c_a) implies higher utility (resp. privacy cost).

We use a function $f : 2^{\Omega} \mapsto [0, 1]$ to measure how much utility vehicle a can achieve given the $\overline{S}_a$ data it obtained, i.e. $h_a = f\left(\overline{S}_a\right)$. f satisfies the following properties:

Property 3.1: (a) $f\left(\overline{S}_a\right) = f\left(\overline{S}_a \cap D_a\right)$, i.e., $\overline{S}_a$ achieves the same utility as its desired part, $\overline{S}_a \cap D_a$;
(b) $f\left(\overline{S}_a\right) = 1$ if $\overline{S}_a \supseteq D_a$, i.e., $\overline{S}_a$ covers all the desired data for vehicle a;
(c) $f\left(\overline{S}_a\right) = 0$ if $\overline{S}_a \cap D_a = \phi$, i.e., $\overline{S}_a$ covers no desired data for vehicle a.
(d) $f\left(\cup_{a \in \mathcal{A}} \overline{S}_a\right) = \sum_{a \in \mathcal{A}} f\left(\overline{S}_a\right)$ if $\left\{\overline{S}_a\right\}_{a \in \mathcal{A}}$ are pairwise disjoint (*countable additivity*).
We use another function $g : 2^{\Omega} \mapsto [0, 1]$ to measure the privacy cost of vehicle a given its shared data set S_a, i.e. $c_a = g\left(S_a\right)$.

According to the lattice-based data-sharing policy, given vehicle a shares its data $S_a^{k_a}$, the vehicle's fitness is calculated by

$$fit_a = \beta \underbrace{xf\left(\cup_{b \in \mathcal{A}, P^{k_b} \subsetneq P^{k_a}} S_b^{k_b}\right)}_{\text{utility}} - \underbrace{g\left(S_a^{k_a}\right)}_{\text{privacy cost}} \tag{1}$$

where the weight $\beta > 0$, called *utility coefficient*, indicates vehicles' willingness to improve their utility. Since each vehicle aims to maximize its own fitness value, we can analyze dynamics of the vehicles' decisions under the data-sharing policy, introduced in Section IV.

IV. Policy Optimization

In this section, we discuss how we optimize the data-sharing policy to converge vehicles' decisions toward the desired decision field. This process includes vehicles' decision dynamics analysis (Section IV-A) and policy optimization itself (Section IV-B).

A. Vehicles' Decision Dynamics Analysis

Analyzing each vehicle's data-sharing decision is non-trivial. Each vehicle might change its decision frequently when it moves from one road segment to another, and road segments may imply different expected levels of data-sharing. Tracking frequent decision changes for each individual will generate an extremely high computation load. To address complexity issues, we analyze vehicles' *group* behaviors, by considering the proportions of different data-sharing decisions taken by vehicles. This also averages out the individual differences (e.g., sensor models, passengers' privacy preferences, and single vehicle's random behaviors) that might also impact vehicles' data-sharing decisions, making the vehicle decision analysis more tractable.

Fig. 4 shows the decision dynamics analysis workflow, which includes 4 steps. Here, we consider vehicles' different decisions for data-sharing in different road segments, quantified by the utility coefficient β in Equ. (1). In what follows, we use the superscript ^{t} to denote the values set/derived in each round t.

Step 1: Road utility coefficient evaluation (Fig. 4(a)): We first evaluate the utility coefficients of different road segments. We partition the road network of the target area into a set of road segments $\mathcal{U} = \{u_1, ..., u_L\}$.

As indicated by [22], a road segment have a higher accident risk when it has higher *betweenness centrality (BC)* or *traffic density (TD)*. Accordingly, we then evaluate the utility coefficient of each road segment via either BC or TD: [22]
1) *BC* of u_i is defined as

$$BC_i = \frac{1}{(N-1)(N-2)} \sum_{u_i, u_j \in \mathcal{R}, u_i \neq u_j \neq u_k} \frac{\eta_{j,k}(u_i)}{\eta_{j,k}}, \tag{2}$$

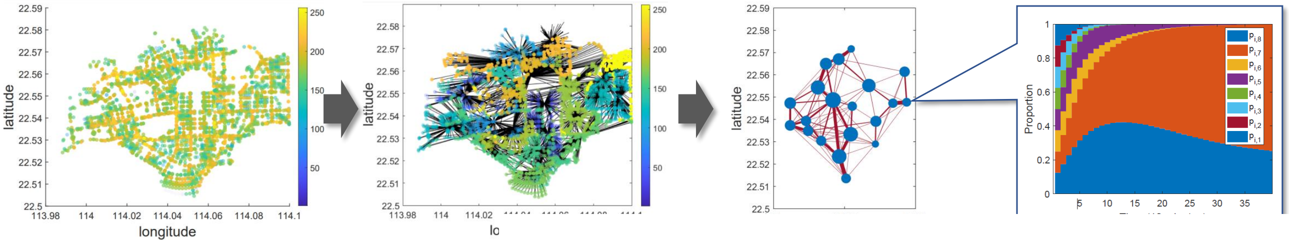
which measures the shortest distance between any two road segments as a way to measure their importance in traversing the network, where $\eta_{i,j}$ is the number of shortest paths between u_j and u_k that contain u_i.
2) *TD* of u_i is defined as

$$TD_i = \frac{\text{\# of vehicles traveling through } u_i \text{ during } [t_s, t_e)}{t_e - t_s}. \tag{3}$$

Step 2: Road segment clustering (Fig. 4(b)): Note that a target area might have over thousands or ten thousand discrete locations (e.g., Futian District in Shenzhen City, China, has around 5,000-6,000 locations), and analyzing vehicles' decisions at each location would lead to an extremely high computation load. On the other hand, we observe that some adjacent locations have similar utility coefficients (betweenness centrality and vehicle traffic). This allows us to develop a coarse-grained model, where we cluster nearby locations with similar utility coefficients to a "region", and approximate the locations' utility coefficients in the same region to a constant. We cluster all the possible locations to M regions r_1, ..., r_M, where the approximated coefficient utility of each region r_i is denoted by β_i^t $(i = 1, ..., M)$.

Here, we apply a heuristic algorithm to cluster the nodes in the road network. The pseudo-code is given in Algorithm 1. The goal of this algorithm is to minimize the variance of node utility coefficients in each cluster so that the error caused by approximation can be minimized.

The algorithm starts by initializing each region r_i with a seed node $r_i = \{u_{\text{seed}_i}\}$ $(i = 1, ..., M)$, where $u_{\text{seed}_1}, ..., u_{\text{seed}_M}$ are evenly distributed over the road network



(a) Utility coefficient evaluation (b) Road s

Fig. 4. Road map of decision dynamics analysis.
*Here we use the vehicle trace data from Shenzhen [21], whi

```
Algorithm 1: Road segment clustering.
Input   : $\mathcal{U}$,
Output  : $r_1, ..., r_M$
1  Select $M$ road segments $u_{\text{seed}_1}, ..., u_{\text{seed}_M}$ in $\mathcal{U}$ that are
     distributed in the area; remove $u_{\text{seed}_1}, ..., u_{\text{seed}_M}$ from
2  Initialize $r_i$ by $\{u_{\text{seed}_i}\}$ $(i = 1, ..., M)$;
3  Initialize $[h_i^{\text{low}}, h_i^{\text{high}}]$ by $[w(u_{\text{seed}_i}), w(u_{\text{seed}_i})]$
     $(i = 1, ..., M)$;
4  Create a queue $\text{queue}_i$ for each $r_i$ $(i = 1, ..., M)$ and pus
     onto $\text{queue}_i$;
5  while $\mathcal{U}$ is nonempty do
6      for each $\text{queue}_i$ $(i = 1, ..., M)$ do
7          Pick up the front node $u$ of $\text{queue}_i$;
8          if $\exists \hat{u} \in \mathcal{N}(u)$ s.t. $w(\hat{u}) \in [h_i^{\text{low}}, h_i^{\text{high}}]$ then
9              Push $\forall \hat{u} \in \mathcal{N}(u)$ s.t. $w(\hat{u}) \in [h_i^{\text{low}}, h_i^{\text{high}}$
                 $\text{queue}_i$;
10             Add $\hat{u}$ to $r_i$;
11             Pop $u$ off $\text{queue}_i$;
12         else
13             Push $\hat{u}$ onto $\text{queue}_i$, where $\hat{u} =$
                 $\arg\min_{u' \in \mathcal{N}(u)} \{|w(u') - h_i^{\text{low}}|, |w(u') - h_i^{\text{high}}|\}$;
14             Add $\hat{u}$ to $r_i$;
15             Update $[h_i^{\text{low}}, h_i^{\text{high}}]$ by
                 $[\min\{h_i^{\text{low}}, w(\hat{u})\}, \max\{h_i^{\text{high}}, w(\hat{u})\}]$;
16 return $r_1, ..., r_M$;
```

(line 1-4). The intuition of the algorithm is to iteratively add the location that is adjacent to each r_i's, and also the difference between the r_i's highest and lowest utility coefficients, h_i^{low} and h_i^{high}, is minimized. Specifically, we use *Breadth-First-Search (BFS)* to traverse the neighbors of r_i (implemented by a queue queue_i), and add the neighbors whose utility coefficients fall in the range of $[h_i^{\text{low}}, h_i^{\text{high}}]$ to r_i (line 10-14). If there exists no such neighbor, the algorithm needs to update the range $[h_i^{\text{low}}, h_i^{\text{high}}]$ by adding the neighbor that increases the range minimally (line 14-16). This process is repeated until all the nodes in $\mathcal{U}$ are added to one of $r_1, ..., r_M$.

Step 3: Graph representation (Fig. 4(c)): As Fig. 5 shows, data-sharing can occur between the vehicles in the same region (a and b), or the vehicles across different regions (c and d). We describe the inter-region data-sharing process using an auxiliary graph $\mathcal{G} = (\mathcal{R}, \mathcal{E})$, where $\mathcal{R} = \{r_1, ..., r_M\}$ and $\mathcal{E} \in \mathcal{R} \times \mathcal{R}$ denote the *node set* and the *edge set* in the graph, respectively. Note that we slightly abuse notation by letting r_i denote both region i and its corresponding node in the auxiliary graph $\mathcal{G}$. If data-sharing can happen between two vehicles in two regions r_i and r_j, then r_i and r_j are *neighbor regions*, and correspondingly, we build an edge $e_{i,j}$ $(e_{i,j} \in \mathcal{E})$ between the two nodes r_i and r_j in $\mathcal{G}$. We let $\mathcal{N}_i$ denote the *neighbor set* of r_i: $\mathcal{N}_i = \{r_j \in \mathcal{R} | \exists e_{i,j} \in \mathcal{E}\}$. We assign a weight $\gamma_{i,j}$ to each $e_{i,j}$ to reflect the data-sharing frequency between the vehicles in r_i and r_j.

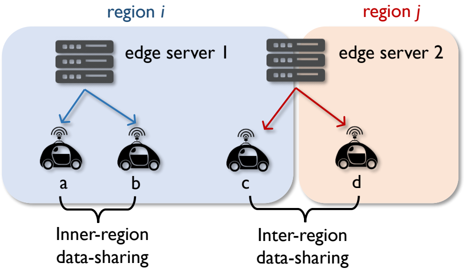


Fig. 5. Inner-region vs. Inter-region data-sharing.
*Edge server 1 covers vehicles a and b, and edge server 2 covers vehicles c and d. vehicles a, b and c are in region i, and vehicle d is in region j.

In each round t, the fitness of vehicles in each region r_i is determined by 1) $\mathbf{p}_i^t$, the decision distribution of vehicles in r_i (*inner-region data-sharing*), 2) $\{\mathbf{p}_j^t\}_{r_j \in \mathcal{N}_i}$, the decision distribution in the neighbor regions $\mathcal{N}_i$ (*inter-region data-sharing*), and 3) $\gamma_{i,i}^t$ and $\{\gamma_{i,j}^t\}_{r_j \in \mathcal{N}_i}$, the *data-sharing frequency* within the region r_i and with the neighbor regions $\mathcal{N}_i$. For simplicity, we use $a \sqsubset r_j$ to represent that vehicle a is in the region r_j. We assume that the shared data $S_a^{l_a}$ from different vehicles are pairwise disjoint. According to *countable additivity* of f (Property 3.1(d)), the fitness of each vehicle selecting decision k is calculated by

$$\begin{aligned} q_{i,k}^t &= \beta_i^t \underbrace{x_i^t \gamma_{i,i}^t f\left(\cup_{a \sqsubset r_i, P^{k_a} \subsetneq P^k} S_a^{k_a}\right)}_{\text{inner-region data-sharing}} \qquad (4) \\ &+ \beta_i^t \underbrace{\textstyle\sum_{r_j \in \mathcal{N}_i} x_j^t \gamma_{j,i}^t f\left(\cup_{a \sqsubset r_j, k_a \preceq k} S_a^{k_a}\right)}_{\text{inter-region data-sharing}} - \underbrace{g\left(S^k\right)}_{\text{privacy cost}} \\ &= \beta_i^t x_i^t \gamma_{i,i}^t \sum_{k_a \preceq k} p_{i,k_a}^t f_{k_a} + \alpha\left(\mathbf{p}_{\mathcal{N}_i,k}^t, \mathbf{x}_{\mathcal{N}_i}^t\right) - g_k. \end{aligned}$$

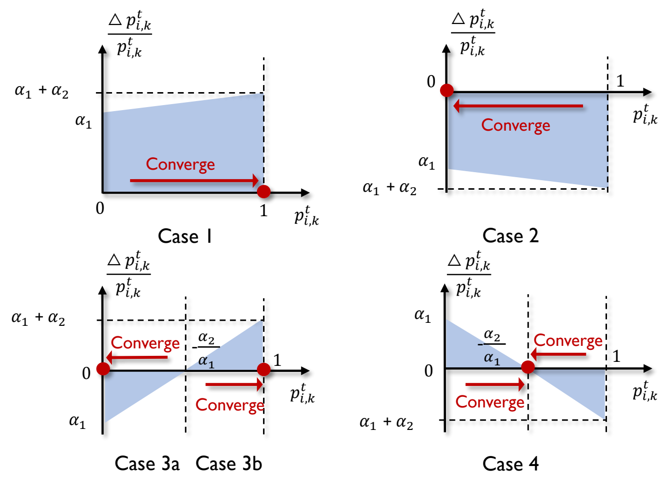


Fig. 6. Different cases of $p^t_{i,k}$'s convergence.

where $\alpha\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) = \beta^t_i \sum_{r_j \in \mathcal{N}_i} x^t_j \gamma^t_{j,i} \sum_{k_a \preceq k} p^t_{j,k_a} f_{k_a}$ denotes the fitness gain contributed by inter-region data-sharing from neighbor regions $\mathcal{N}_i$.

Step 4. Decision dynamics analysis (Fig. 4(d)): After obtaining the fitness value $p^t_{i,k}$ of vehicle selecting decision k in Equ. (4), $p^t_{i,k}$'s per capita growth rate, $\frac{\triangle p^t_{i,k}}{p^t_{i,k}}$ $(\triangle p^t_{i,k} = p^{t+1}_{i,k} - p^t_{i,k})$ can be calculated by the replicator dynamics [23].

$$\frac{\triangle p^t_{i,k}}{p^t_{i,k}} = q^t_{i,k} - \overline{q}^t_i = \alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) p^t_{i,k} + \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \tag{5}$$

where $\overline{q}^t_i = \sum_{l=1}^{K} p^t_{i,l} q^t_{i,l}$ is the average utility of all in vehicles in region r_i at round t, and

$$\alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) = g_k - \beta^t_i x^t_i \gamma^t_{i,i} \sum_{k_a \prec k} p^t_{i,k_a} f_{k_a} - \alpha\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)$$

$$\begin{aligned} &\alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) = \alpha\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \\ +\; & \beta^t_i x^t_i \gamma^t_{i,i} \left(\sum_{k_a \prec k} p^t_{i,k_a} f_{k_a} - \sum_{l=1, l\neq k}^{K} p^t_{i,l} \sum_{k_a \preceq l, k_a \neq k} p^t_{i,k_a} f_{k_a} \right) \\ +\; & \sum_{l=1, l\neq k}^{K} g_l p^t_{i,l} - g_k - \sum_{l=1, l\neq k}^{K} p^t_{i,l} \alpha\left(\mathbf{p}^t_{\mathcal{N}_i,l}, \mathbf{x}^t_{\mathcal{N}_i}\right) \end{aligned}$$

According to the replicator dynamics in Equ. (5), given different sharing-ratio $\mathbf{x}$, the convergence of $p^t_{i,k}$ can be categorized into the following 4 cases, as shown in Fig. 6:

Case 1: When $\mathbf{x}^t \in \mathcal{X}^t_{i,1}$, $p^t_{i,k}$ converges to 1, where

$$\mathcal{X}^t_{i,1}\left(\mathbf{x}^t\right) = \left\{ \mathbf{x}^t \;\middle|\; \begin{array}{l} \alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) + \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \geq 0, \\ \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \geq 0 \end{array} \right\}. \tag{6}$$

Case 2: When $\mathbf{x}^t \in \mathcal{X}^t_{i,2}$, $p^t_{i,k}$ converges to 0, where

$$\mathcal{X}^t_{i,2}\left(\mathbf{x}^t\right) = \left\{ \mathbf{x}^t \;\middle|\; \begin{array}{l} \alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) + \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \leq 0, \\ \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \leq 0 \end{array} \right\}. \tag{7}$$

Case 3a: When $\mathbf{x}^t \in \mathcal{X}^t_{i,3\mathrm{a}}$, $p^t_{i,k}$ converges to 0, where

$$\mathcal{X}^t_{i,3\mathrm{a}}\left(\mathbf{x}^t\right) = \left\{ \mathbf{x}^t \;\middle|\; \begin{array}{l} \alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) + \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \geq 0, \\ \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \leq 0, \\ p^t_{i,k} \geq -\frac{\alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)}{\alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)} \end{array} \right\}. \tag{8}$$

Case 3b: When $\mathbf{x}^t \in \mathcal{X}^t_{i,3\mathrm{b}}$, $p^t_{i,k}$ converges to 1, where

$$\mathcal{X}^t_{i,4}\left(\mathbf{x}^t\right) = \left\{ \mathbf{x}^t \;\middle|\; \begin{array}{l} \alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) + \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \geq 0, \\ \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \leq 0, \\ p^t_{i,k} \leq -\frac{\alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)}{\alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)} \end{array} \right\}. \tag{9}$$

In **Case 3a&b**, the rest point $p^t_{i,k} = -\frac{\alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)}{\alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)}$ is *unstable*.

Case 4: When $\mathbf{x}^t \in \mathcal{X}^4_i$, $p^t_{i,k}$ converges to $-\frac{\alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)}{\alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)}$, called the *Evolutionarily stable strategies (ESS)*, where

$$\mathcal{X}^t_{i,4}\left(\mathbf{x}^t\right) = \left\{ \mathbf{x}^t \;\middle|\; \begin{array}{l} \alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) + \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \leq 0, \\ \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) \geq 0, \\ -\frac{\alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)}{\alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)} \in \mathcal{P}^*_{i,k} \end{array} \right\}. \tag{10}$$

B. Policy Optimization

We use $\mathcal{P}^*_{i,k}$ to denote the desired decision field of each $p^t_{i,k}$ $(i = 1, ..., M, k = 1, ..., K)$. The objective of policy optimization is to find an optimal sharing ratio vector $\mathbf{x}^t = [x^t_1, ..., x^t_M]$ in each round t such that each $p^t_{i,k}$ converges to its desired decision field $\mathcal{P}^*_{i,k}$ as fast as possible. We use t_0 to denote the first round when the policy optimization is applied, and $\tilde{t}$ to denote the first round when each $p^t_{i,k}$ converges to the desired decision field, i.e.,

$$p^{\tilde{t}}_{i,k} \in \mathcal{P}^*_{i,k}, \;\; \forall i = 1, ..., M, k = 1, ..., K \tag{11}$$

According to the replicator dynamics in Equ. (5), we can obtain

$$p^{\tilde{t}}_{i,k} = \sum_{t=t_0}^{\tilde{t}-1} \left(\alpha_1\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right) p^t_{i,k} + \alpha_2\left(\mathbf{p}^t_{\mathcal{N}_i,k}, \mathbf{x}^t_{\mathcal{N}_i}\right)\right) p^t_{i,k} + p^{t_0}_{i,k} \tag{12}$$

On the other hand, we aim to change $\mathbf{x}$ smoothly by setting a constraint for the change of x_i in each round, i.e.,

$$x^{t+1}_i - x^t_i \leq \Lambda, \;\; t = t_0, ..., \tilde{t} - 1 \tag{13}$$

Given the desired decision fields $\left\{\mathcal{P}^*_{i,k}\right\}_{i=1,...,M, k=1,...,K}$, we formally formulate the policy optimization problem as:

$$\min\ \tilde{t} \quad \text{s.t.} \quad \text{Equ. (11) (12) (13) are satisfied,} \tag{14}$$

where $\{\mathbf{x}^t\}_{t=t_0,\ldots,\tilde{t}-1}$ are the decision variables.

The above problem is a non-convex optimization problem, which is NP-hard in general [24]. Due to its computational intractability, in the following we propose a time-efficient distributed algorithm, namely the ***Fast Decision Shaping (FDS)*** algorithm, that can effectively converge $p_{i,k}^t$ to its desired decision field with low time complexity.

FDS Algorithm. According to the decision dynamics analysis in Section IV-A, the high-level idea of FDS is to relocate the rest point for each $p_{i,k}^t$ such that $p_{i,k}^t$ is in the area flowing to the desired decision field. Algorithm 2 shows the pseudo-code.

Algorithm 2: The FDS algorithm.

```
   Input   : $\left\{\mathcal{P}_{i,k}^*\right\}_{i=1,\ldots,M,k=1,\ldots,K}$, $\mathbf{x}^{t_0} = \left[x_1^{t_0},\ldots,x_M^{t_0}\right]$
   Output  : $\mathbf{x}^{t_0}, \mathbf{x}^{t_0+1}, \ldots, \mathbf{x}^t$
1  $t \leftarrow t_0$;
2  while $\exists p_{i,k}^t \notin \mathcal{P}_{i,k}^*$ do
3      for each $i = 1,\ldots,M$ do
4          for each $k = 1,\ldots,K$ do
5              case $1 \in \mathcal{P}_{i,k}^*$ do
6                  $\mathcal{X}_{i,k}^t \leftarrow \mathcal{X}_{i,1}^t \cup \mathcal{X}_{i,3a}^t$;
7              case $0 \in \mathcal{P}_{i,k}^*$ do
8                  $\mathcal{X}_{i,k}^t \leftarrow \mathcal{X}_{i,2}^t \cup \mathcal{X}_{i,3b}^t$;
9              case $0,1 \notin \mathcal{P}_{i,k}^*$ do
10                 $\mathcal{X}_{i,k}^t \leftarrow \mathcal{X}_{i,4}^t$;
11         $\mathcal{X}_i^t \leftarrow \cap_{k=1}^K \mathcal{X}_{i,k}^t$;
12         if $x_i^t \in \mathcal{X}_i^t$ then
13             $x_i^{t+1} \leftarrow x_i^t$;
14         else
15             if $x_i^t \leq \min\left\{\mathcal{X}_i^t\right\}$ then
16                 $x_i^{t+1} \leftarrow x_i^t + \min\left\{\Delta, \min\left\{\mathcal{X}_i^t\right\} - x_i^t\right\}$;
17             else
18                 $x_i^{t+1} \leftarrow x_i^t - \min\left\{\Delta, x_i^t - \min\left\{\mathcal{X}_i^t\right\}\right\}$;
19     $t \leftarrow t + 1$;
20 return $\mathbf{x}^{t_0}, \mathbf{x}^{t_0+1}, \ldots, \mathbf{x}^t$;
```

The algorithm iteratively derives the value of each x_i until all $p_{i,k}^t$ converges to its desired decision field. In each round t, the cloud servers determines x_i considering the following cases (**Case 1–4** in Section IV-A):

1) When $1 \in \mathcal{P}_{i,k}^*$ (line 5-6), a sufficient condition to converge $p_{i,k}^t$ to the desired decision field (or 1) is to let

$$\begin{aligned} & x_i^t \in \mathcal{X}_{i,1}^t\left(x \left| x_j^t = x_j^{t-1}, \forall j \text{ s.t. } j \neq i\right.\right) && (15)\\ \text{or} \quad & x_i^t \in \mathcal{X}_{i,\text{3a}}^t\left(x \left| x_j^t = x_j^{t-1}, \forall j \text{ s.t. } j \neq i\right.\right), && (16)\end{aligned}$$

such that **Case 1** or **Case 3a** can be achieved.

2) When $0 \in \mathcal{P}_{i,k}^*$ (line 7-8), a sufficient condition to converge $p_{i,k}^t$ to the desired decision field (or 0) is to let $\mathbf{x} \in \mathcal{X}_{i,2}^t \cup \mathcal{X}_{i,\text{3b}}^t$.

$$\begin{aligned} & x_i^t \in \mathcal{X}_{i,2}^t\left(x \left| x_j^t = x_j^{t-1}, \forall j \text{ s.t. } j \neq i\right.\right) && (17)\\ \text{or} \quad & x_i^t \in \mathcal{X}_{i,\text{3b}}^t\left(x \left| x_j^t = x_j^{t-1}, \forall j \text{ s.t. } j \neq i\right.\right), && (18)\end{aligned}$$

such that **Case 2** or **Case 3b** can be achieved.

3) When $0,1 \notin \mathcal{P}_{i,k}^*$ (line 9-10), $p_{i,k}^t$ convergences to the desired decision field essentially implies to relocate the rest point $p_{i,k}^*$ (stable ESS), so that $p_{i,k}^t$ is in the area that flows to the desired decision field. Hence, a sufficient condition to converge $p_{i,k}^t$ to the desired decision field (or 1) is to let

$$x_i^t \in \mathcal{X}_{i,4}^t\left(x \left| x_j^t = x_j^{t-1}, \forall j \text{ s.t. } j \neq i\right.\right), \tag{19}$$

such that **Case 4** can be achieved.

After finding $\mathcal{X}_i^t$, intersection of the conditions to converge $\mathbf{x}$ to the desired decision field (line 11), the algorithm checks whether x_i^t satisfies the condition $\mathcal{X}_i^t$. If yes, then the current $\mathbf{x}^t$ in the policy is good, $\mathbf{x}^t$ remains the same in the next round (line 12-13); otherwise, $\mathbf{x}^t$ is moved towards to the desired decision field with the maximum step (line 14-18).

Lower bound of the optimal solution. For theoretical interest, we also design an algorithm to derive the lower bound of the privacy optimization problem (Equ. (14)), to check how close FDS can achieve to the optimal.

Proposition 4.1: Given $\mathbf{p}_i^t$ and x_i^t, each $\triangle p_{i,k}^t$ is bounded by

$$\begin{aligned} \triangle p_{i,k}^t \quad & \leq \quad \beta_i\left(1 - p_{i,k}^t\right) \sum_{k_a \preceq k} f_{k_a}\left(\gamma_{i,i} x_i + \sum_{r_j \in \mathcal{N}_i} \gamma_{j,i}\right) p_{i,k}^t \\ & - \quad \left(g_k - \sum_{l=1}^K p_{i,l}^t g_l\right) p_{i,k}^t && (20)\\ \triangle p_{i,k}^t \quad & \geq \quad -\beta_i \sum_{l=1, l \neq k}^K p_{i,l}^t \sum_{k_a \preceq l} f_{k_a}\left(\gamma_{i,i} x_i + \sum_{r_j \in \mathcal{N}_i} \gamma_{j,i}\right) p_{i,k}^t \\ & - \quad \left(g_k - \sum_{l=1}^K p_{i,l}^t g_l\right) p_{i,k}^t && (21)\end{aligned}$$

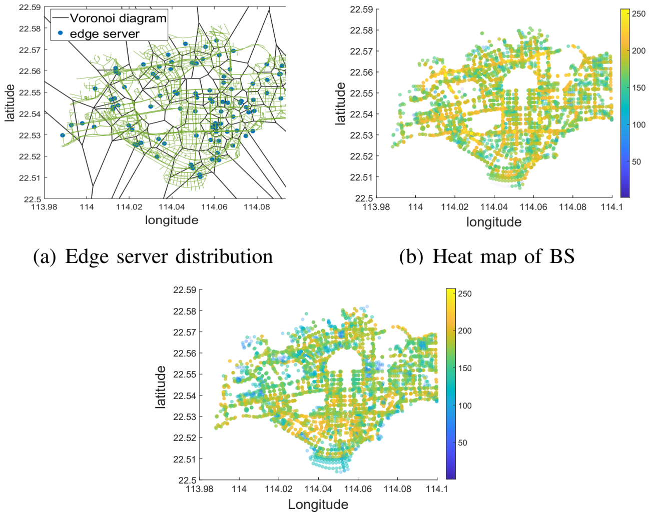
Proof The detailed proof can be found in Appendix.

According to Proposition 4.1, by relaxing the constraint of Equ. (12) to Equ. (20) and Equ. (21), we can formulate the following relaxed policy optimization problem

$$\min\ \tilde{t} \quad \text{s.t.} \quad \text{Equ. (11) (13) (20) (21) are satisfied} \tag{22}$$

of which the constraints in Equ. (11) (13) are linear, and the constraints in Equ. (20) (21) are quadratic, indicating the problem's feasible region is convex.

To find the optimal solution of the relaxed problem, we initialize the value of $\tilde{t}$ by 1, and check whether there exists a feasible solution satisfying the constraints in Equ. (11)(13)(20)(21). As the feasible region is convex, feasibility checking can be implemented via the subgradient method [24]. If no feasible solution exists, $\tilde{t}$ is increased by 1, and we check the feasibility again. This process is repeated until a feasible solution is found, and the corresponding $\tilde{t}$ is a lower bound of the solution of the original policy optimization problem.

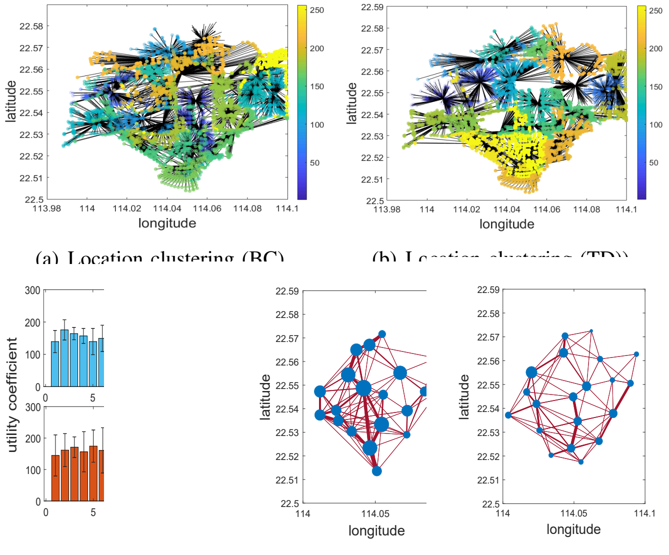


(a) Edge server distribution

(b) Heat map of BS

(c) Heat map of average TD

Fig. 7. Vehicle dataset.



(a) Location clustering (BC)

(b) Location clustering (TD)

(c) β in di

(d) Graph (BS)

(e) Graph (TD)

Fig. 8. Region partition (clustering).

V. Performance Evaluation

In this section, we carry out a trace-driven simulation to evaluate the performance of our data-sharing policy. We also provide results on the impact of the distribution of vehicles' data-sharing decisions toward convergence, and convergence speed.

A. Dataset and experiment settings

The dataset adopted in the simulation contains the timestamps, GPS positions, and velocities of around 27,996 vehicles in Shenzhen, China [21], including 15,610 taxicabs and 12,386 customized transit service vehicles in Dada Car corporation[1]. Here, we use taxicabs and transit service vehicles as a proxy of vehicles by assuming that vehicles follow similar mobility patterns with taxicabs and transit service vehicles in the road network.

We select *Futian district* in Shenzhen as our target area. The road map information is obtained from OpenStreetMap [25]. According to the municipal information of Futian, we use a bounding box with coordinate *(latitude = 22.50, longitude = 113.98)* as the south-west corner, and coordinate *(latitude = 22.59, longitude = 114.10)* as the north-east corner to crop the road map data.

As Figure 7(a) shows, 100 stationary edge servers are evenly deployed in the target area. Given the positions of the edge servers, the whole target area is partitioned into 100 Voronoi cells. In every 10 seconds, each vehicle reports its collected sensor data to the edge server located within the same cell, which is the nearest edge server of this vehicle. We set each round as 10 minutes.

We calculate the *betweenness centrality* (BC for short, defined in Equ. (2)) and the *traffic density* (TD for short, defined in Equ. (3)) of all the road segments in the target area, and depict their heat maps in Fig. 7(b) and Fig. 7(c), respectively. To calculate TD, we first count TD for each road segment (i.e. the number of vehicles traveling through the road segment) every 10 minutes and calculate the average value of TD over one day as the utility coefficient for each road segment.

B. Road segment clustering

Given the road segment utility coefficients calculated by BC and TD, we apply Algorithm 1 to cluster the road segments in the target area into 20 regions, as shown in Fig. 8(a) and Fig. 8(b), respectively. Fig. 8(c) shows the distributions of the utility coefficients in different regions, where the bar height represents the average coefficient value, and the interval includes 95% coefficient values within the region. The average standard deviations of utility coefficients are respectively 17.08 and 30.31 when the coefficients are calculated by BC and TD, implying that TD has a higher standard deviation within the regions. This is because when clustering, TD is calculated based on its average value over the whole time span (one day). So, two road segments in the same region might have similar average TD over a time period, but their TD at each time point might have a higher difference.

Fig. 8(d) and Fig. 8(e) show the graph presentations of the clustered regions calculated by BC and TD, respectively. In the figures, each node represents a region, and its size represents the number of road segments within the region. The width of each edge represents the average inter-region communication frequency between the corresponding two regions in one day.

C. Convergence of the vehicles' data-sharing decision

Given the partitioned regions, we now test how the proportion of the different data-sharing decisions converges over time with our data-sharing policy.

In Table II, we quantify the utility and the privacy cost of the eight data-sharing decisions. The utility of each decision is calculated on the basis of a recent survey [19], which summarizes how much utility LiDAR, radar, and camera contribute to the 11 factors of vehicles' perception, listed in

[1] In the simulation, the identities of all the vehicles have been removed.

Table III. The contributions of different sensors to each facto can be in one of three levels: *"competently"*, *"reasonabl well"*, and *"doesn't operate well"*, and are quantified by 1, 0. and 0, respectively. Then, the utility of a decision is equal t the sum contribution of its shared sensor data to the 11 factor For instance, the utility of D7 = {Camera, LiDAR} is equal t "13", the sum of "7" and "6", which are the sum contribution of camera and LiDAR to the 11 factors.

TABLE II
PRIVACY COST.

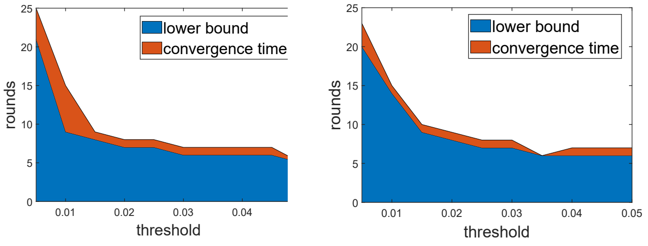
Privacy policies	Utility	Privacy cost
$P^1 = \{\Omega_{\text{camera}}, \Omega_{\text{lidar}}, \Omega_{\text{radar}}\}$	20	1.6
$P^2 = \{\Omega_{\text{camera}}, \Omega_{\text{lidar}}\}$	13	1.5
$P^3 = \{\Omega_{\text{camera}}, \Omega_{\text{radar}}\}$	14	1.1
$P^4 = \{\Omega_{\text{lidar}}, \Omega_{\text{radar}}\}$	13	0.6
$P^5 = \{\Omega_{\text{camera}}\}$	7	1.0
$P^6 = \{\Omega_{\text{lidar}}\}$	6	0.5
$P^7 = \{\Omega_{\text{radar}}\}$	7	0.1
$P^8 = \{\}$	0	0

TABLE III
UTILITY CONTRIBUTION OF DIFFERENT SENSORS IN PERCEPTION.

Factors	Camera	LiDAR	Radar
Range	0.5	0.5	1
Resolution	1	0.5	0
Distance Accuracy	0.5	1	1
Velocity	0.5	0	1
Color perception, e.g., traffic lights	1	0	0
Object detection	0.5	1	1
Object classification	1	0.5	1
Lane detection	1	0	0
Obstacle edge detection	1	1	0
Illumination conditions	0	1	1
Weather conditions	0	0.5	1
Sum contribution to the 11 factors	7	6	7

We quantify the privacy cost of the three types of sensor data by relying on the vehicles' sensor privacy survey [17], as well as on the use of those sensors in inference attacks [8]. Overall, we rank camera data as "*highest sensitive*", LiDAR data as "*moderate sensitive*", and radar data as "*least sensitive*", and their privacy costs are quantified by 1.0, 0.5, and 0.1, respectively. The privacy cost of a decision is defined as the sum privacy cost of its shared sensor data. For instance, the privacy cost of D7 = {Camera, LiDAR} is "1.5", which is the sum privacy cost of camera (1.0) and LiDAR data (0.5). Finally, we normalize both utility and privacy cost to the range of [0, 1].

Based on the above settings, we run the simulation and evaluate the convergence time of FDS (Algorithm 2), i.e., the time duration that vehicles' data-sharing decisions converges to the desired decision field, given different desired decision fields. For instance, under the weather such as fog, rain and snow, we require a higher proportion of camera information (e.g., $p_{i,1} = p_{i,2} = p_{i,3} = p_{i,5} = 20\%$, $p_{i,8} = 20\%$, and all the others are 0%) in the desired decision field, while on a sunny day, the proportion of camera data is set lower (e.g., $p_{i,1} = 65\%$, $p_{i,5} = 25\%$, $p_{i,7} = p_{i,8} = 5\%$, and all the others are 0%). Moreover, we allow an acceptable error ϵ for the desired decision field. That is, given a desired value $p^*_{i,k}$ for $p_{i,k}$, the convergence time of $p_{i,k}$ is the time duration that $p_{i,k}$ converges to the interval $\left[p^*_{i,k} - \epsilon, p^*_{i,k} + \epsilon\right]$.



(a) Convergence time (BS) (b) Convergence time (TD)

Fig. 9. Convergence time of FDS.

Note that, the desired decision field $\mathcal{P}^*_{i,k}$ of each $p_{i,k}$ can be changed given different environmental factors, e.g., weather, traffic, etc. The goal of the experiment is to test how fast vehicles' decisions can converge to a given desired decision field. The derivation of the desired decision fields given various environmental factors is out of the scope of this paper and will be studied in our future work.

Fig. 9(a) and Fig. 9(b) show the convergence time of FDS with ϵ increased from 0.01 to 0.05, given that the utility coefficients are calculated by BC and TD, respectively. Not surprisingly, the convergence time decreases with the increase of ϵ in both figures, i.e., the convergence time is shortened when the desired decision fields are loosened. Particularly, if the acceptable error is increased to 2%, the convergence time can drop immediately (e.g., to 7 or 8 rounds).

Moreover, we compare the convergence time of FDS with the theoretical lower bound of convergence time, derived by solving the relaxed optimal policy problem defined in Equ. (22). Note that the optimal convergence time is located in the gap between the convergence time of FDS and the lower bound of the minimum convergence time. The results demonstrate that our approach can achieve a point close to the optimal solution, where the approximation ratios are in the range of [1.00, 1.15] and [1.00, 1.08], when the utility coefficients are defined by BS and TD, respectively.

Finally, to take a closer look at how the vehicles' data-sharing decisions change over time under our policy, we pick up a desired field: $p^*_{i,1} = 65\%, p^*_{i,5} = 25\%, p^*_{i,7} = p^*_{i,8} = 5\%$ and $p^*_{i,2} = p^*_{i,3} = p^*_{i,4} = p^*_{i,6} = 0\%$, and depict the change of the proportion of data-sharing decisions with and without the sharing ratio controlled by FDS in Fig. 10. We first set the sharing ratio x_i by two constants 0.2 and 1.0, without the control of FDS. Not surprisingly, when $x_i = 0.2$, the vehicles' decisions converge to "sharing no data" ($p_{i,8} = 13\%$) or "only LiDAR data" ($p_{i,7} = 87\%$), since the low sharing ratio discourages the data-sharing among vehicles. In contrast, when $x_i = 1.0$, the edge servers forward the collected data with full ratio, incentivizing vehicles' to "share all data" ($p_{i,1} = 76\%$) or "share camera data" ($p_{i,5} = 24\%$). Under neither cases, the vehicles' decisions converge to the desired field.

The third figure shows that FDS converges the proportion of different decisions to the desired field: $p^*_{i,1} = 65\%, p^*_{i,5} = 25\%, p^*_{i,7} = p^*_{i,8} = 5\%$. The fourth figure shows the proportion difference of decisions in adjacent rounds, from which we find that the convergence speed is fast in the first 8 rounds,

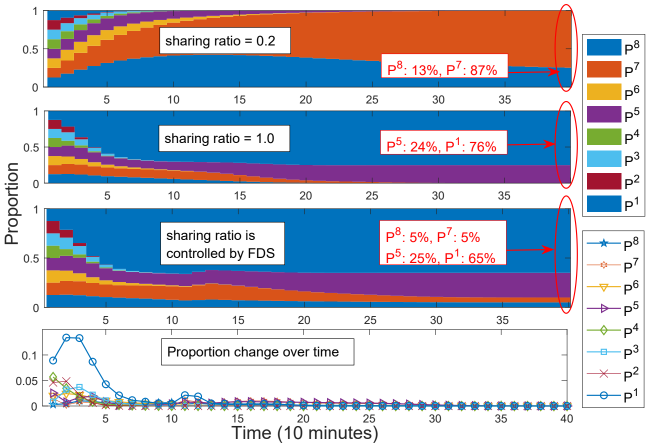


Fig. 10. Convergence of the population of different data-sharing decisions.

but after that, there is a long tail of convergence. The results are consistent with the observations in Fig. 9(a)(b), i.e., if the acceptable error is increased slightly (e.g., by 2%, the convergence time can drop quickly (e.g., to 5 rounds).

VI. Related Work

Vehicles' cooperative perception. The past few years have witnessed the rapid development of vehicles, particularly their technologies to perceive surrounding obstacles [26]–[28]. Data fusion on multi-sensors has been well investigated to facilitate the development of 3D object detection [29]–[31]. For example, Labayrade et al. [31] developed a low-level sensor data fusion to extract the features or objects for object tracking. Kaempchen et al. [32] developed a scalable feature-level sensor fusion architecture, which combines multi-layer data of laser-scanner and monocular video for the purpose of object tracking. Xu et al. [33] proposed 3D object detection methods by fusing both image and point cloud from the same vehicle. While elegant, all of those works collect sensor data from individual vehicles, of which the perception accuracy is limited by the coverage of their own sensors.

Recently, a rich body of works has applied cooperative perception to improve vehicles' perception accuracy. Rauch et al. [26] first establish the foundation of cooperative perception by developing a high-level sensor data fusion architecture, called Car2X-based perception, which delivers a vehicle's consistent results for fusion with the results generated by the host vehicle. Qiu et al. [34] proposed to crowdsource sensing tasks to various vehicles to provide wider spatial coverage as well as disambiguation. Instead of using high-level sensor data, Chen et al. [3], [4] proposed to fuse original calibrated raw LiDAR data from multiple vehicles to improve 3D detection precision in a low-level data fusion method.

Privacy protection of connected vehicles. Data sharing among vehicles have raised many privacy and safety concerns. In fact, the privacy issues of connected vehicles (not limited to autonomous vehicles) have been studied by a rich body of recent literature (not limited to autonomous vehicles) [10]–[12]. Some of those works apply traditional privacy protection techniques like cryptography [13] and *anonymity* [14], [35], while others focused on *obfuscation* [11], [12], wherein which drivers are allowed to share perturbed information instead of exact information to servers.

Although privacy protection techniques so far have not received much attention in cooperative perception compared with conventional connected vehicles, several recent works have started investigating the privacy issues for connected autonomous vehicles. For example, Selena et al. [9] investigated the public opinion about automated and connected vehicles in Australia and New Zealand, particularly people's concerns about different types of sensors in vehicles. Bloom et al. investigated people's (e.g. pedestrian's) privacy concerns for their surrounding autonomous vehicles [17]. While, to the best of our knowledge, this paper is the first work that aims to address the privacy issue of AVs in cooperative perception via a policy-driven strategy.

VII. Conclusions

In this paper, we proposed a new data-sharing policy for the cooperative perception of autonomous vehicles. Considering the different data-sharing desires of vehicles under different scenarios, our policy provides vehicles autonomy to determine what types of sensor data to share. Furthermore, by analyzing the dynamics of vehicles' data-sharing decisions, we designed a policy optimization strategy to dynamically adjust data-sharing ratios among vehicles, such that vehicles' decision can converge to the desired decision fields, and hence a healthy cooperation environment for vehicles can be maintained in a long term. The experimental results from the trace-driven simulation have demonstrated the effectiveness and efficiency of our policy.

We envision several promising directions to continue this research. First, we will study the desired field of view of individual vehicle given the vehicle's surrounding environment as well as its primary goals (route planning, collision avoidance, etc), so that we can estimate the utilities of different data-sharing decisions more accurately. Second, we will further consider the case that edge servers can perceive their surrounding environment as well and distribute their own perception to the bypassed vehicles. Since vehicles usually move at a relatively high speed, the time duration that vehicles connect to their edge servers is limited. To this end, we will study how to distribute sensor data from edge servers to vehicles in a time-efficient manner. Finally, we will take into account the dynamic desire of individual vehicles' data-sharing decisions, e.g., how vehicles might change their decision from peak hours to off-peak hours, and analyze how the approximation errors of utility coefficients might impact the convergence time of vehicles' decisions.

VIII. Acknowledgements

This work was partly supported by U.S. NSF grants CNS-2029881, CNS-2029976, CNS-2037982, CNS-2113805, CNS-1852134, OAC-2017564, Industrial application research project of Shenzhen for undertaking the national key project of China (No. CJGJZD20210408091600002), Science and Technology Development Fund of Macau SAR, China (File no.

0015/2019/AKP), and Basic Research Foundation of Guangdong, China (No. 2020B1515130004).

Appendix

Proof of Proposition 4.1

First, for each decision k in the region r_i, the utility gain obtained from inter-region communication $\alpha\left(\mathbf{p}_{\mathcal{N}_i,k}, \mathbf{x}_{\mathcal{N}_i}\right)$ can be bounded by:

$$0 \leq \alpha\left(\mathbf{p}_{\mathcal{N}_i,k}, \mathbf{x}_{\mathcal{N}_i}\right) \leq \beta_i \textstyle\sum_{r_j \in \mathcal{N}_i} \gamma_{j,i} \sum_{k_a \preceq k} f_{k_a}.$$

According to which, we can find a upper bound of each $\frac{\triangle p_{i,k}^t}{p_{i,k}^t}$:

$$\begin{aligned}
\frac{\triangle p_{i,k}^t}{p_{i,k}^t} &\leq \beta_i x_i \gamma_{i,i} \left(\sum_{k_a \preceq k} p_{i,k_a}^t f_{k_a} - \sum_{l=1}^{K} p_{i,l}^t \sum_{k_a \preceq l} p_{i,k_a}^t f_{k_a} \right) \\
&+ \beta_i \left(1 - p_{i,k}^t\right) \sum_{r_j \in \mathcal{N}_i} \gamma_{j,i} \sum_{k_a \preceq k} f_{k_a} - \left(g_k - \sum_{l=1}^{K} p_{i,l}^t g_l \right) \\
&\leq \beta_i \left(1 - p_{i,k}^t\right) \sum_{k_a \preceq k} f_{k_a} \left(\gamma_{i,i} x_i + \sum_{r_j \in \mathcal{N}_i} \gamma_{j,i} \right) \\
&- \left(g_k - \sum_{l=1}^{K} p_{i,l}^t g_l \right),
\end{aligned}$$

and a lower bound

$$\begin{aligned}
\frac{\triangle p_{i,k}^t}{p_{i,k}^t} &\geq \beta_i x_i \gamma_{i,i} \left(\sum_{k_a \preceq k} p_{i,k_a}^t f_{k_a} - \sum_{l=1}^{K} p_{i,l}^t \sum_{k_a \preceq l} p_{i,k_a}^t f_{k_a} \right) \\
&- \sum_{l=1, l \neq k}^{K} p_{i,l}^t \beta_i \sum_{r_j \in \mathcal{N}_i} \gamma_{j,i} \sum_{k_a \preceq l} f_{k_a} - \left(g_k - \sum_{l=1}^{K} p_{i,l}^t g_l \right) \\
&\geq -\beta_i \sum_{l=1, l \neq k}^{K} p_{i,l}^t \sum_{k_a \preceq l} f_{k_a} \left(\gamma_{i,i} x_i + \sum_{r_j \in \mathcal{N}_i} \gamma_{j,i} \right) \\
&- \left(g_k - \sum_{l=1}^{K} p_{i,l}^t g_l \right).
\end{aligned}$$

The proof is completed.

References

[1] "Waymo," https://waymo.com/, 2022, accessed: 2022-01-011.

[2] "Tusimple," https://www.tusimple.com/, 2022, accessed: 2022-01-011.

[3] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in *Proc. of ICDCS*, 2019, pp. 514–524.

[4] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds," in *Proc. of SEC*, 2019, p. 88–100.

[5] S. Biswas, R. Tatchikou, and F. Dion, "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety," *IEEE Communications Magazine*, vol. 44, no. 1, pp. 74–82, 2006.

[6] J. Gozalvez, M. Sepulcre, and R. Bauza, "Ieee 802.11p vehicle to infrastructure communications in urban environments," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 176–183, 2012.

[7] S. Rangarajan, M. Verma, A. Kannan, A. Sharma, and I. Schoen, "V2c: A secure vehicle to cloud framework for virtualized and on-demand service provisioning," in *Proc. of the ICACCI*, 2012, p. 148–154.

[8] "The privacy implications of autonomous vehicles," Downloaded from https://www.dataprotectionreport.com/2017/07/the-privacy-implications-of-autonomous-vehicles/, 2017.

[9] "Public opinion about automated and connected vehicles in australia and new zealand: Results from the 2nd advi public opinion survey," https://www.yelp.com/, 2020, accessed: 2020-04-07.

[10] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-vehicle network attacks and countermeasures: Challenges and future directions," *IEEE Network*, vol. 31, no. 5, pp. 50–58, 2017.

[11] C. Qiu, A. C. Squicciarini, Z. Li, C. Pang, and L. Yan, "Time-efficient geo-obfuscation to protect worker location privacy over road networks in spatial crowdsourcing," in *Proc. of ACM CIKM*, 2020.

[12] C. Qiu, A. C. Squicciarini, C. Pang, N. Wang, and B. Wu, "Location privacy protection in vehicle-based spatial crowdsourcing via geo-indistinguishability," *IEEE TMC*, pp. 1–1, 2020.

[13] Q. Wu, J. Domingo-Ferrer, and U. Gonzalez-Nicolas, "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 559–573, 2010.

[14] L. Zheng, H. Yue, Z. Li, X. Pan, M. Wu, and F. Yang, "k-anonymity location privacy algorithm based on clustering," *IEEE Access*, 2018.

[15] C. Qiu and A. C. Squicciarini, "Location privacy protection in vehicle-based spatial crowdsourcing via geo-indistinguishability," in *Proc. of IEEE ICDCS*, 2019, pp. 1061–1071.

[16] M. Cominelli, F. Kosterhon, F. Gringoli, R. L. Cigno, and A. Asadi, "An experimental study of csi management to preserve location privacy," in *Proc. of WiNTECH*. ACM, 2020, p. 64–71.

[17] C. Bloom, J. Tan, J. Ramjohn, and L. Bauer, "Self-driving cars and data collection: Privacy perceptions of networked autonomous vehicles," in *Proc. of SOUPS*. USENIX, Jul. 2017, pp. 357–375.

[18] C. Qiu, H. Shen, and K. Chen, "An energy-efficient and distributed cooperation mechanism for k-coverage hole detection and healing in wsns," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1247–1259, 2018.

[19] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, 2021.

[20] H. Li, L. Liu, C. Lan, C. Wang, and H. Guo, "Lattice-based privacy-preserving and forward-secure cloud storage public auditing scheme," *IEEE Access*, vol. 8, pp. 86 797–86 809, 2020.

[21] L. Yan and et al., "Catcharger: Deploying in-motion wireless chargers in a metropolitan road network via categorization and clustering of vehicle traffic," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[22] X. Wang, X. Wu, and M. Abdel-Aty, "Investigation of road network features and safety performance." *Accident; analysis and prevention*, 2013.

[23] C. Qiu and et al., "Combating behavioral deviance via user behavior control," in *Proc. of AAMAS*, 2018, p. 202–210.

[24] F. S. Hillier, *Linear and Nonlinear Programming*. Stanford University, 2008.

[25] "openstreetmap," https://www.openstreetmap.org/, 2020-04-07.

[26] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer, "Car2x-based perception in a high-level fusion architecture for cooperative perception systems," in *2012 IEEE IVS*, 2012, pp. 270–275.

[27] J. R. et al., "Accurate single stage detector using recurrent rolling convolution," 2017.

[28] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," 2017.

[29] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," 2018.

[30] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," 2020.

[31] R. Labayrade, C. Royere, D. Gruyer, and D. Aubert, "Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner," *Auton. Robots*, vol. 19, no. 2, p. 117–140, sep 2005.

[32] N. Kaempchen, M. Buehler, and K. Dietmayer, "Feature-level fusion for free-form object tracking using laserscanner and video," in *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, 2005, pp. 453–458.

[33] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," 2018.

[34] H. Qiu, J. Chen, S. Jain, Y. Jiang, M. McCartney, G. Kar, F. Bai, D. K. Grimm, M. Gruteser, and R. Govindan, "Towards robust vehicular context sensing," *IEEE TVT*, vol. 67, no. 3, pp. 1909–1922, 2018.

[35] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 571–588, 2002.