sensors

MDPI

Article

Decentralized Policy Coordination in Mobile Sensing with Consensual Communication

Bolei Zhang [1,2], Lifa Wu [1] and Ilsun You [3,*]

[1] School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
[2] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China
[3] Department of Financial Information Security, Kookmin University, Seoul 02707, Republic of Korea
* Correspondence: isyou@kookmin.ac.kr

Abstract: In a typical mobile-sensing scenario, multiple autonomous vehicles cooperatively navigate to maximize the spatial–temporal coverage of the environment. However, as each vehicle can only make decentralized navigation decisions based on limited local observations, it is still a critical challenge to coordinate the vehicles for cooperation in an open, dynamic environment. In this paper, we propose a novel framework that incorporates consensual communication in multi-agent reinforcement learning for cooperative mobile sensing. At each step, the vehicles first learn to communicate with each other, and then, based on the received messages from others, navigate. Through communication, the decentralized vehicles can share information to break through the dilemma of local observation. Moreover, we utilize mutual information as a regularizer to promote consensus among the vehicles. The mutual information can enforce positive correlation between the navigation policy and the communication message, and therefore implicitly coordinate the decentralized policies. The convergence of this regularized algorithm can be proved theoretically under certain mild assumptions. In the experiments, we show that our algorithm is scalable and can converge very fast during training phase. It also outperforms other baselines significantly in the execution phase. The results validate that consensual communication plays very important role in coordinating the behaviors of decentralized vehicles.

Keywords: mobile sensing; reinforcement learning; decentralized coordination; communication

Citation: Zhang, B.; Wu, L.; You, I. Decentralized Policy Coordination in Mobile Sensing with Consensual Communication. *Sensors* **2022**, *22*, 9584. https://doi.org/10.3390/s22249584

Academic Editor: Felipe Jiménez

Received: 24 October 2022
Accepted: 4 December 2022
Published: 7 December 2022

1. Introduction

Over the past decade, the ubiquitous adoption of mobile vehicles has greatly enhanced the flexibility and convenience of environment sensing. When equipped with sensors, multiple vehicles can autonomously navigate to different locations to collect distributed environmental data. This paradigm, often referred to as mobile sensing, has attracted attention from a variety of disciplines, such as air quality sensing [1], traffic monitoring [2], fire detection [3], etc. For example, in a smart home, multiple devices (e.g., sweeping robots) can cooperate to sense the environment and perform related tasks [4], such as cleaning and tidying.

In a typical mobile sensing scenario, multiple events (e.g., fire, traffic jam, and pollution emission) may occur randomly and dynamically (depicted in Figure 1). Detecting such events in time is crucial for the mobile sensing application. However, since each vehicle can only observe the local environment within a limited radius, one central problem emerging is **how to navigate the decentralized vehicles to maximize the spatial–temporal coverage of the events**. As the vehicles need to make sequential navigation decisions, reinforcement learning (RL), in particular, multi-agent reinforcement learning (MARL) methods, have become a promising approach. RL methods can be model-free to optimize the navigation policies through exploration and exploitation. They are, therefore, applicable in different scenarios, even when the environmental model is not assumed [5,6].

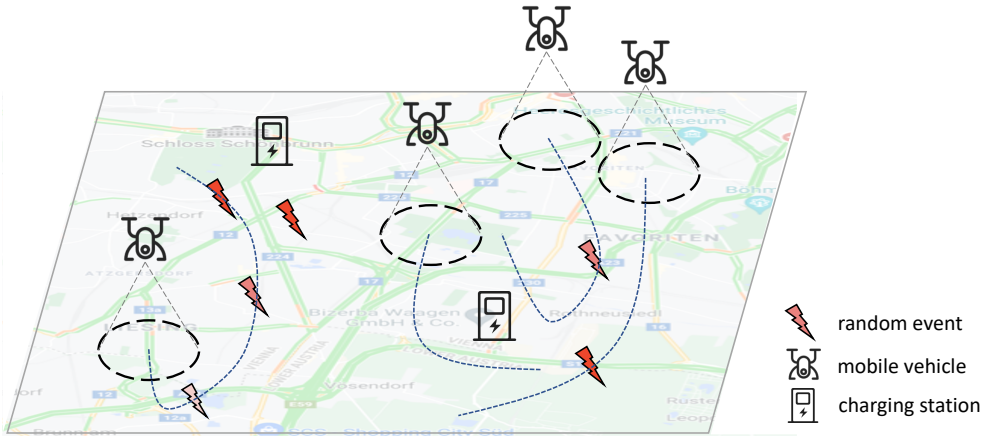

Figure 1. An illustration of the mobile sensing, where multiple vehicles cooperate to monitor the random events. The blue dash lines represent the moving trajectory of each vehicle. Events with a redder color imply higher intensities.

Despite the progress made in recent years, one critical challenge that has been largely overlooked is the decentralized coordination of the vehicles. As illustrated in Figure 1, the events are mostly distributed at the left and right sides of the map. It could be better if one of the right vehicles moves to the left area for sensing. However, without coordination, the right vehicles may compete to sense nearby events, leading to wasted sensing efforts. One possible direction to tackle this challenge is to use a centralized controller that manages the policies of all vehicles. However, centralized approaches may face the problem of "single point of failure" and low scalability.

To navigate multiple vehicles in an open, dynamic environment, we adopt the MARL as the basic solution. However, in the execution phase, the vehicles may still have uncoordinated behaviors due to the lack of common consensus [7–9]. Inspired by the recent advances of learning to communicate [10,11], we can also introduce the communication mechanism in the cooperative navigation. On one hand, the common signal can provide global information from all the vehicles. On the other hand, the other vehicles' moving actions can also be inferred if there is positive correlation with the signal.

Our Method In this paper, we consider the decentralized management of the mobile vehicles, and introduce a communication-based framework to coordinate the behaviors of the vehicles. At each step before moving, the vehicles should first broadcast communication messages to others to share information. Afterwards, when receiving the communication messages from others, each vehicle can be conditioned on the received messages to take navigation actions. By adopting this communication framework, the vehicles can share information with each other to break through the dilemma of local observation. In particular, the communication message is also learned via reinforcement learning with the aim to maximize the spatial–temporal coverage of the events. This learning to communicate framework is flexible, and can be applicable in different dynamic environments.

One major concern in the communication framework is that the vehicles may simply ignore the communication message and focus only on local observations. To deal with this problem, in this paper, we try to **maximize the mutual information between the received messages and the vehicles' navigation policies**. By maximizing this term, the mobile vehicles can correlate their policies with the received messages. Intuitively, a positive correlation implies that other vehicles' policies can be inferred based on the received message. Therefore, the vehicles can achieve consensus implicitly. Theoretical analysis shows that this regularized algorithm can converge to equilibrium points under certain mild conditions.

In the experiment part, we implement and evaluate the proposed algorithm in a simulation environment built from a real-world data set. We first validate the decentralized algorithm in both the training and execution phases. The results show that the consensual

communication framework can successfully coordinate the behaviors of the decentralized vehicles. The mutual information term plays an important role in the coordination. Our method can also adapt to multiple scenarios with different hyper-parameters. In different settings, our algorithm can consistently outperform other baselines. Our work can be widely adopted in different fields, such as smart homes, smart city, agriculture, etc.

1.1. Contributions

Our key contributions are listed as follows:

- We model the mobile sensing problem as a decentralized sequential optimization problem, where the vehicles navigate to maximize the spatial-temporal coverage of the events in the environment.
- A communication framework is proposed for cooperative navigation. In particular, the communication protocol is learned by model-free reinforcement learning methods.
- We explicitly correlate the vehicles moving policies with the communication messages to promote coordination. The regularized algorithm can be proved to converge to equilibrium points under certain mild assumptions.
- Extensive experiments are conducted to show the effectiveness of our approach.

1.2. Organizations

The rest of the paper is organized as follows. We first introduce the related work in Section 2. Next, we formulate the system model and the optimization objective in Section 3. Section 4.1 presents the framework of learning to communicate. We then present how to enforce positive communication in Section 5. Evaluation is given in Section 6. We conclude the paper in Section 7.

2. Related Work

In this section, we first introduce the recent advances in reinforcement learning, which is the main technical solution in this work. Next, we will review the related works of mobile sensing, with a focus on how to navigate the mobile vehicles in the environment to maximize the event coverage.

2.1. Reinforcement Learning

Reinforcement learning (RL) has achieved great success in wide areas, such as Game of Go [12], Atari [13], Starcraft [8], etc. The problem of RL can generally be modeled as a Markov decision process (MDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, and $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition model for generating the next state. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. $\gamma \in (0, 1]$ is a discount factor. At each step t, when an agent observes the state $s^t \in \mathcal{S}$ and executes an action $a^t \in \mathcal{A}$, it will then be transitioned into a new state s^{t+1} and receive an immediate reward r^t, with probability $p(s^{t+1}, r^t \mid s^t, a^t) \in \mathcal{T}$. Let R^t denote the cumulative return at time t. In an infinite horizon MDP, the cumulative return can be represented as

$$R^t = \lim_{h \to \infty} \frac{1}{h} \sum_{t'=t}^{h} \mathbb{E}[r^{t'}] \tag{1}$$

The goal of reinforcement learning is to find the optimal policy μ^* to maximize the return: $\mu^* = \arg\max_\mu \mathbb{E}_\mu[R^0]$, where policy $\mu(a_t \mid s_t)$ is a function which maps the state s_t to a distribution of actions a_t. MDP has the property of the Bellman equality:

$$Q(s^t, a^t) = r^t + \gamma \sum_{s^{t+1}} p(s^{t+1} \mid s^t, a^t) v(s^{t+1}) \tag{2}$$

where $Q(s, a) = \mathbb{E}[R^t \mid s^t = s, a^t = a]$ is the state-action value function and $v(s) = \mathbb{E}[R^t \mid s^t = s]$ is the value function of state s.

The process of RL can be generally divided into training and execution phases. In the training phase, the RL agent uses exploration and exploitation in the environment to optimize the policy. While in the execution phase, the agent will fix the policy parameters in the environment. In this paper, as the vehicles need to move in a continuous space, we focus on DDPG [14,15], which can generate continuous actions. In DDPG, there is a critic function to evaluate the state-action value by following a deterministic policy μ as $Q^{\mu}(s^t, a^t)$, and an actor function which maps the state s^t to a deterministic action, $a^t = \mu(s^t)$.

Recently, multi-agent reinforcement learning (MARL) has also been a hot research topic. MARL models the environment as a decentralized partially observable Markov decision process (Dec-POMDP) [8,9] as a tuple $\langle \mathcal{S}, \mathcal{T}, \mathcal{A}, \mathcal{R}, \mathcal{O}, \mathcal{I} \rangle$, where $\mathcal{O}$ is the set of local observations and $\mathcal{I}$ is the set of agents. The agents that make decisions are based on the observations. Let $o_i^t \in \mathcal{O} \subseteq \mathcal{S}$ be the local observation of agent i at step t. Each agent i can choose an action $a_i^t \in \mathcal{A}$, forming a joint action $\mathbf{a}^t \in \mathcal{A}^n$, and transition to the next state $s^{t+1} \in \mathcal{S}$ according to the function $p(s^{t+1}, r^t \mid s^t, \mathbf{a}^t) \in \mathcal{T}$, where the reward function $r^t \in \mathcal{R}$ is shared by all the agents.

To optimize the policies of the agents in MARL, previous works, such as COMA [8], MADDPG [9], QMIX [7], etc., mainly adopted the "centralized training, decentralized execution" (CTDE) mechanism: during training, global state information can be used to train the policy network; and during execution, the agents can only condition on local observations. In the execution phase, the agents could still change their policies dynamically, leading to incoordination of the decentralized policies. However, we address that such a CTDE mechanism may not be applicable in decentralized environments where the agents can only be trained separately. Recent works are considering methods of learning to communicate [16–19], where the communication policy is learned via RL. We will also adopt this mechanism in our work. In comparison to previous works [16,17,19] that mostly use lazy communication, we propose to enforce positive communication so that the messages can be utilized more efficiently. Moreover, most of previous works only used ungrounded, cheap talk for communication [10]. We address that such cheap talk communication may not be effective in coordination.

2.2. Mobile Sensing

Mobile sensing has been extensively studied with the emergency of autonomous vehicles. One of the main problem is maximizing the coverage of events in the environment. Earlier works mostly assumed that the environment model is a prior and proposed combinatorial optimization method. For example, Karaliopoulos et al. [20] modeled the problem as a cover problem and proposed the approximation ratio algorithm. Hu et al. [21] also proposed mobile sensing methods with spatial–temporal awareness. The paper adopted a combinatorial pinning zero-determinant (ZD) strategy to find a cost-efficient mobile sensing strategy. In comparison, our work addresses the dynamics of the environment, and the coordinated policies of different mobile users are learned via repeated interactions.

As the users make independent decisions, decentralized algorithms based on game theory were also considered. Rahili et al. [22] designed a rule-based communication protocol in which agents can communicate with local neighbors and use their local information make decisions. Esch et al. [23] depicted a distributed algorithm where the agents can communicate with one another wirelessly within a fixed communication radius. Li et al. [24] modeled the mobile crowdsourcing as a Stackelberge game, and proposed a three-party evolutionary game model for task allocation. However, most previous methods are hard to generalize to unseen scenarios. In an open environment, it is critical for the agents to adapt to dynamic environment events. Data privacy is also important in mobile sensing and has been a hot research topic very recently [25–29]. In comparison, we focus more on the navigation of the mobile vehicles instead of the data-collecting process.

When the environment model is unknown, machine learning approaches attract attention [30,31]. In particular, as the environment is often dynamic [32,33], online learning or RL-based algorithms are widely considered, which are sequential and model-free. An

et al. [34] adopted the multi-armed bandits method to select users to improve service quality. However, bandit algorithms neglect the sequential behavior of agents and may not be feasible for mobile sensing problems. As RL uses deep learning to extract the representation of the environment for exploration and exploitation, it can be naturally applicable in the dynamic environment. For example, Zhang et al. [35] adopted RL for a coarse-to-fine deep scheme to address the aspect ratio variation in UAV tracking. Liu et al. [36,37] used deep RL for high quality data collection. The main idea is to employ multiple mobile vehicles to schedule their paths independently to maximize the coverage of distributed POIs (point of interests). Zeng et al. [38] divided the problem into four sub-optimal problems, and used an iterative algorithm solve the optimal problem. Liu et al. [5] proposed a multi-UAV mobile sensing framework based on multi-agent reinforcement learning (MARL), and utilized "centralized training decentralized execution" (CTDE) for cooperation. Wei et al. [6] considered the multi-robot informative path planning problem and proposed independent learning through credit assignment for cooperative sensing. Samir et al. [39] leveraged unmanned aerial vehicles (UAVs) for mobile sensing and proposed an RL approach to maximize the sensing coverage. A major challenge in these works is to coordinate the policies of different mobile vehicles for cooperation. While most previous works implicitly learn the cooperation policies for each agent, in our work, we addressed that coordination is crucial and explicitly proposed policy coordination methods based on consensual communication.

3. System Model

In this paper, we consider a mobile sensing problem where a set of mobile vehicles $\mathcal{I} = \{1, 2, \ldots, N\}$ cooperate to maximize the spatial–temporal sensing coverage of the events in the environment. Suppose the time horizon is divided into infinite discrete intervals as $\{0, 1, 2, \ldots, \infty\}$. At each interval t, each vehicle $i \in \mathcal{I}$ at position (x_i^t, y_i^t) need to decide the moving action a_i^t, which can be represented as a tuple of speed $v_i^t \in [0, S_{\max}]$ and angle $\phi_i^t \in [0, 2\pi)$, i.e., $a_i^t = (v_i^t, \phi_i^t)$. After moving, the new position will be $(x_i^t + v_i^t \sin\phi_i^t, y_i^t + v_i^t \cos\phi_i^t)$. Meanwhile, vehicle i is associated with a battery capacity $b_i^t \in [0, b_{\max}]$. The battery has a consumption rate Δ_i^t that is linear with the vehicle speed, i.e., $\Delta_i^t = \beta v_i^t + \Delta_0$, where β is a coefficient and Δ_0 is a constant intrinsic battery consumption. The battery capacity will be updated as $b_i^{t+1} = b_i^t - \Delta_i^t$ each time. To avoid running out of power, the vehicles should regularly move to the charging station, in which the battery will be recharged for a fixed number of units b_0 at each interval.

In the environment, random events may happen at different positions with time-varying intensities. Let $\mathcal{E}$ be the set of events. We use τ_e^t, $e \in \mathcal{E}$ to represent the event intensity of e at step t. The event e at position (x_e^t, y_e^t) is sensed/covered by vehicle i if it is within a limited radius of i. Let $\mathbb{1}_{ie}^t$ be an indicator function to represent if the event is covered by vehicle i:

$$\mathbb{1}_{ie}^t = \begin{cases} 1, & \text{if } \sqrt{(x_i^t - x_e^t)^2 + (y_i^t - y_e^t)^2} \le l_i, \\ 0, & \text{o.w.} \end{cases} \tag{3}$$

where l_i is the sensing radius of vehicle i. The benefit will be τ_e^t if the event e is covered by at least one of the mobile vehicles. Note that if multiple vehicles cover the same event e simultaneously, the benefit is still τ_e^t. Therefore, the vehicles should cooperate to avoid repeatedly sensing the same event. We use $\mathbb{1}_e^t$ as an indicator function that the event e is covered by at least one vehicle at interval t, i.e., $\mathbb{1}_e^t = \max\{\mathbb{1}_{1e}^t, \mathbb{1}_{2e}^t, \ldots, \mathbb{1}_{Ne}^t\}$. The problem can then be formulated as finding the joint moving policies for the vehicles, so that the cumulative sensing coverage of the events is maximized:

$$\begin{aligned} &\max \sum_{t=0}^{\infty} \sum_{e \in \mathcal{E}} \tau_e^t \mathbb{1}_e^t \\ &\text{s.t. } b_i^t \ge 0, \forall i \in \mathcal{I}, \forall t \in \{0, 1, 2, \ldots \infty\}. \end{aligned} \tag{4}$$

The inequality constraint in the objective means that the mobile vehicles could no longer move or sense when running out of battery. According to the objective, the vehicles need to make sequential navigation decisions to cover the dynamic events. However, as the vehicles make decentralized decisions, it could be difficult for the vehicles to acknowledge others' observations or intentions. This brings the dilemma of local observation and will be the main focus of this paper. Table 1 summarizes the key parameters in this paper.

Table 1. Key parameter table of system model.

Notation	Definition
$\mathcal{I}$	the set of mobile vehicles: $\mathcal{I} = \{1, 2, \ldots, N\}$
$\mathcal{E}$	the set of events
t	the time step $t \in \{0, 1, \ldots, \infty\}$
(x_i^t, y_i^t)	the position of vehicle i at step t
(v_i^t, ϕ_i^t)	moving speed and angle of vehicle i at step t
b_i^t	battery capacity of vehicle i at step t
Δ_i^t	battery consumption rate of vehicle i at step t
b_0	battery charging rate at the charging station
l_i	the sensing radius of vehicle i
τ_e^t	the event intensity of event $e \in \mathcal{E}$ at step t
c	the penalty for running out of battery
r_i^t	the reward of vehicle i at step t
$\mu_i(\cdot)$	the moving policy function of vehicle i
$\mu_i^m(\cdot)$	the communication policy function of vehicle i
$Q_i(\cdot)$	the action–value function of vehicle i
$v_i(\cdot)$	the state value function of vehicle i
$q(\cdot)$	proxy for the posterior function
$I(\cdot)$	mutual information function
ρ	the weight of the MI reward

4. Learning to Communicate

To break through the dilemma of local observation, in this section, we first formulate the problem as a Markov game. Then we formally introduce the communication framework, where the vehicles can share information with each other. Finally, we will show how to optimize the moving policies of each vehicle under this framework.

4.1. Mobile Sensing as a Markov Game

According to the system model, we can formulate the mobile sensing problem as a Dec-POMDP with tuples of $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{I} \rangle$, where the set of agents $\mathcal{I}$ represent the mobile vehicles. Now we give the definitions of other elements as follows:

- *State*: In the mobile sensing problem, at each interval t, the system state $s^t \in \mathcal{S}$ includes the global information of the environment.
- *Observation*: In the environment, each vehicle i can only partially observe the state. The observation $o_i^t \in \mathcal{O}$ is the subset of the environment state: $o_i^t \subseteq s^t$. We assume that each vehicle can observe the environment information within the sensing radius l_i, including its own position, last moving action, remaining battery capacity and sensed events.

- *Action*: The action of the mobile vehicle i is a continuous tuple $a_i = (v_i^t, \phi_i^t) \in \mathcal{A}_i$, where v_i^t is the speed, and ϕ_i^t represents the moving angle. At each interval, all the vehicles will take the moving action to form a joint action $\mathbf{a}^t = (a_1^t, a_2^t, \ldots, a_N^t)$.
- *Transition*: Given the joint actions of the vehicles, the environment will transit to the next state s^{t+1} according to the transition function:

$$p(s^{t+1} \mid s^t, \mathbf{a}^t) : \mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \to \mathcal{S} \tag{5}$$

  Note that this function is not known to be used, and can only be inferred through repeated interaction with the environment.
- *Reward*: As the mobile vehicles cooperate to maximize the spatial–temporal coverage of the environment, we define a global reward as the sensed events intensities:

$$r^t = \sum_{e \in \mathcal{E}} \tau_e^t \mathbb{1}_e^t \tag{6}$$

  However, for each vehicle, it is intricate to infer its contribution to the global reward. Therefore, we decompose the reward function and define the individual reward for each vehicle i as

$$r_i^t = \sum_{e \in \mathcal{E}, \mathbb{1}_{ie}^t = 1} \frac{\tau_e^t \mathbb{1}_{ie}^t}{\sum_{k \in \mathcal{I}} \mathbb{1}_{ke}^t} \tag{7}$$

  The reward function indicates that the reward of sensing event e is averaged by the number of vehicles that cover e at this step. It is obvious to see that $r^t = \sum_{i \in \mathcal{I}} r_i^t$. To take the battery capacity into account, we relax the constraint in Equation (4) with an additional term c when the vehicle runs out of battery power. The vehicles will receive this penalty when the capacity is below zero, i.e., $c(b_i^t) = c$ if $b_i^t < 0$; otherwise, $c(b_i^t) = 0$. The value of c balances the preference between sensing a reward and penalty of battery loss. The relaxed version of the reward can be formulated as

$$r_i^t = \sum_{e \in \mathcal{E}, \mathbb{1}_{ie}^t = 1} \frac{\tau_e^t \mathbb{1}_{ie}^t}{\sum_{k \in \mathcal{I}} \mathbb{1}_{ke}^t} - c(b_i^t) \tag{8}$$

4.2. The Communication Framework

As the vehicles only have limited observation, we introduce a communication framework to share information among the vehicles. Figure 2 presents an illustration of the communication procedure. We now separately describe how to broadcast and receive the messages.

Communication Broadcasting As presented in Figure 2, at each step t before moving, each vehicle i first broadcasts a message m_i^t to other vehicles. When broadcasting the message, an intuitive idea is to send the observation o_i^t and the intended action a_i^t to other vehicles. However, this is not possible since the vehicle will also be conditioned on the received messages from others to take action a_i^t. Moreover, the dimensions of the observation may be large with high overhead. Instead, we introduce the mechanism of **learning to communicate**. Suppose vehicle i uses a communication policy network $\mu_i^m(o_i^t)$ parameterized by θ_i^m to output the message content m_i, which can be a fixed-size continuous vector. In particular, the communication policy network can be optimized via the RL-based algorithm, where the goal is the long-term cumulative sensing coverage of the events. By learning to communicate, the vehicles can encode the observations and intentions into a compact embedding, which significantly reduces the transmission cost. Moreover, it can be flexible to deal with different scenarios and environments. More details on how to optimize the communication policy network will be introduced in Section 4.3.

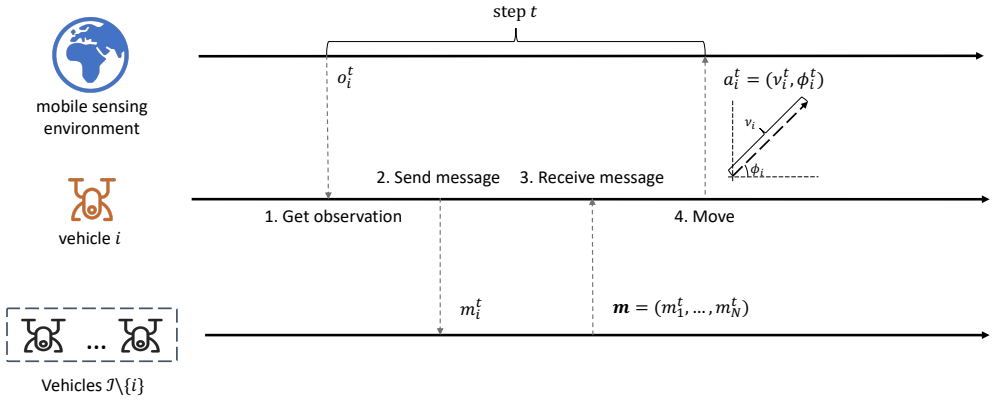

Figure 2. An illustration of the procedure. With local observation, the vehicles first exchange their messages by broadcasting. Afterwards, they make the moving decision based on the received message and local observation. This figure shows the communication process of the ego vehicle i in a single step t: The vehicle first obtains the observation o_i^t in the environment. The vehicle then broadcasts a message m_i^t to other vehicles based on the observation. The messages $(m_1^t, \ldots, m_N^t)$ from other vehicles will also be aggregated as part of the observation. Vehicle i will finally make the moving decision a_i^t based on the environment observation and the aggregated message.

Communication Receiving After broadcasting, each vehicle can also receive the messages from other vehicles: $\mathbf{m}^t = (m_1^t, m_2^t, \ldots, m_N^t)$. The messages can be aggregated with different operators, such as mean, max, or neural networks, such as recurrent neural networks (RNN). The aggregated message can be represented as $m_g = \text{AGG}(\mathbf{m}^t)$, where AGG is the aggregator of the received messages. Suppose the moving policy of vehicle is represented as $\mu_i(\cdot)$. It can be formulated as conditioning on the local observation and received messages for moving: $a_i^t = \mu_i^t(o_i^t, m_g)$.

4.3. Policy Optimization

With the communication framework, we can now optimize the moving policy networks $\mu_i(\cdot)$ and communication policy networks $\mu_i^m(\cdot)$ for each vehicle $i \in \mathcal{I}$. As the moving action of each vehicle is a continuous vector, we use DDPG for policy optimization. Let $Q_i(\cdot)$ be the action value function (critic) parameterized by θ_i^Q. (We temporarily abbreviate the time indicator t. The sign $^-$ indicates $t-1$ and $'$ indicates $t+1$.) The policy functions $\mu_i^m(\cdot)$, $\mu_i(\cdot)$ and the critic function $Q_i(\cdot)$ can all be implemented with neural networks. The parameters θ_i of the moving policy network $\mu_i(\cdot)$ can be updated according to the deterministic policy gradient theorem [14]:

$$\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{o_i, \mathbf{m}, a_i \sim \mathcal{D}}[\nabla_{\theta_i} \mu_i(o_i, m_g) \nabla_{a_i} Q_i(o_i, a_i, m_g) \mid_{a_i = \mu_i(o_i, m_g)}] \tag{9}$$

where $J(\cdot)$ is the return of the policy and $\mathcal{D}$ is the set of historical data samples. Similarly, we can also update the parameters of the communication policy network μ_i^m as

$$\nabla_{\theta_i^m} J(\mu_i^m) = \mathbb{E}_{o_i, \mathbf{m}, a_i \sim \mathcal{D}}[\nabla_{\theta_i^m} \mu_i^m(o_i) \nabla_{m_i} Q_i(o_i, a_i, m_g) \mid_{m_i = \mu_i^m(o_i)}] \tag{10}$$

where θ_i^m represents the parameters of the communication policy network. The action value network can be updated by minimizing the temporal difference (TD) error:

$$\begin{aligned} \mathcal{L}(\theta_i^Q) = & \mathbb{E}_{o_i, m_g, a_i, r_i, o_i', m_g'}[Q_i(o_i, a_i, m_g) - \\ & (r_i + \gamma Q_i(o_i', a_i', m_g') \mid_{m_i' = \mu_i^m(o_i'), a_i' = \mu_i(o_i', m_g')}]^2 \end{aligned} \tag{11}$$

According to the above formulations, we can update the parameters of the policy networks and action value networks simultaneously. Compared to the CTDE framework, which requires centralized training, in our framework, the networks can be optimized

independently based on the local observation and communication messages. Therefore, this framework can be applicable in decentralized training scenarios.

5. Consensual Communication

By learning to communicate, the mobile vehicles can share local information with each other. However, previous works have shown that selfish agents do not learn to use this type of ungrounded, cheap talk communication channel effectively [11]. In this section, we first try to enforce the mobile vehicles to have **consensual communication**, i.e., the communication will indeed influence the vehicles' behaviors. Next, we show that the algorithm can converge under the communication framework.

5.1. Mutual Information for Consensual Communication

To enforce positive communication, we maximize the **mutual information** between the moving policy μ_i and the aggregated message from i's neighbors: m_g. Intuitively, by maximizing the mutual information, the vehicle can correlate its moving policy with the messages from neighbors. This can also be regarded as reducing the uncertainty of vehicles' moving policy after receiving the messages. Formally, we augment the reward function as follows:

$$\hat{r}_i = (1-\rho) r_i + \rho I(\mu_i; m_g) \tag{12}$$

where $\rho \in [0, 1]$ is a hyper-parameter that controls the importance of the mutual information term $I(\mu_i; m_g)$. The mutual information item can be expressed in terms of entropy and conditional entropy:

$$\begin{aligned} I(\mu_i; m_g) &= \mathcal{H}(m_g) - \mathcal{H}(m_g \mid \mu_i) \\ &= \mathcal{H}(\mu_i) - \mathcal{H}(\mu_i \mid m_g) \end{aligned} \tag{13}$$

where $\mathcal{H}(\cdot)$ is the entropy function. The mutual information will become zero if the communication message does not influence the moving policy. In this case, $\mathcal{H}(m_g)$ equals $\mathcal{H}(m_g \mid \mu_i)$. Maximizing the mutual information indicates that *we enforce all the vehicles to correlate their policies with the message*. Thus, the vehicles can infer other neighbors' behaviors by acknowledging the broadcast message, which implicitly promotes coordination among the vehicles. However, directly maximizing the MI is intractable. We instead introduce the variational distribution $q(m_g \mid \mu_i)$ as a proxy for the posterior over μ_i. Learning a neural network to predict the messages based on the policy μ_i provides a lower bound on MI:

$$\begin{aligned} &I(\mu_i; m_g) = \mathcal{H}(m_g) - \mathcal{H}(m_g \mid \mu_i) \\ &= \mathcal{H}(m_g) + \mathbb{E}_{m_g, \mu_i}[\log q(m_g \mid \mu_i)] \\ &\quad - \mathbb{E}_{m_g}[D_{KL}(p(m_g \mid \mu_i) \mid\mid q(m_g \mid \mu_i))] \\ &\geq \mathcal{H}(m_g) + \mathbb{E}_{\mu_i, m_g}[\log(q(m_g \mid \mu_i)] \end{aligned} \tag{14}$$

where D_{KL} is the KL divergence between two probabilities. The establishment of inequality is because the KL-divergence distance is non-negative. In practice, as the policy μ_i is a network, we use historical observation–action trajectories $traj_i$ to represent the policy.

The network structure of our framework is presented in Figure 3. For each vehicle, there are four neural networks associated, including one critic network, two actor networks, and an additional variation network which is used for policy coordination. The output of the critic network can be used to update the actor networks during training. For the variation network, even though the gradient cannot be backpropagated to update the actor–critic networks, the augmented reward function can guide the mobile vehicles to generate coordinated behaviors. In the network structures, FC means fully connected, and GRU is gated recurrent unit. GRU is used to extract information from the sequential observations. More details of the network parameters will be introduced in the experiment part. As the network parameters for each vehicle can be optimized in a decentralized way, this framework can be scalable to a large number of mobile vehicles.

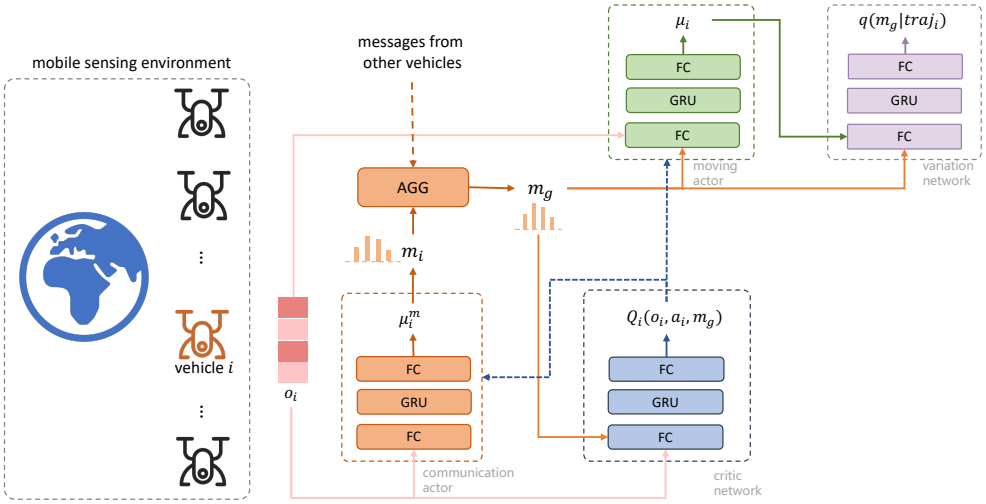

Figure 3. The network structure of policy coordination with consensual communication. FC represents fully connected layer, and GRU represents gated recurrent unit layer. AGG is the aggregator operator for the received messages. According to the network structure, each mobile vehicle needs to maintain 4 networks: moving actor network (the green part), critic network (the blue part), communication actor network (the orange part) and the variation network (the purple part).

Algorithm Now we formally present the algorithm in Algorithm 1 for an ego vehicle i. In this algorithm, we first initialize the parameters of the networks for the ego agent i. At each step, we generate the broadcast message m_i based on the current observation o_i. The agent will then receive and aggregate messages from others and execute actions a_i. The tuples will be stored into the replay buffer $\mathcal{D}$. During training, we sample a mini-batch of tuples from the buffer and perform gradient back propagation to update the critic network and actor networks. Finally, the variation network is also trained by maximizing the mutual information.

Algorithm 1: Policy optimization for ego vehicle i

Randomly initialize neural networks Q_i, μ_i^m, μ_i;
Initialize global replay buffer $\mathcal{D} = \emptyset$;
Initialize message $\mathbf{m} \leftarrow 0$, random processes $\mathcal{N}_i^m$ and $\mathcal{N}_i^a$;
Receive observations of vehicle i o_i;
while *not converge* **do**
  /* Execute actions */
  Broadcast message $m_i \leftarrow \mu_i^m(o_i) + \mathcal{N}_i^m$;
  Receive messages $m_g = \mathrm{AGG}(\mathbf{m})$;
  Execute moving action $a_i \leftarrow \mu_i(o_i, m_g) + \mathcal{N}_i^a$;
  Get reward r_i and next observation o_i';
  Store $\langle o_i, a_i, \mathbf{m}, o_i', r_i \rangle$ in $\mathcal{D}$;
  Set $o_i = o_i'$;
  /* Train networks */
  Sample K tuples $(o_i, a_i, \mathbf{m}, o_i', r_i, \mathbf{m}')$ from $\mathcal{D}$;
  Update moving actor $\mu_i(\cdot)$ with Equation (9);
  Update critic network $Q_i(\cdot)$ with Equation (11);
  Update communication actor $\mu_i^m(\cdot)$ with Equation (10);
  Update variation network $q(\cdot)$ by maximizing Equation (14);
end

Complexity According to the above algorithm, we give a formal analysis of the time complexity of the training phase for each ego vehicle. At each step of training, the vehicle need to sample K tuples and update the networks. The update of the networks takes

$O(1)$ complexity for gradient descent. Suppose the convergence takes C steps. The time complexity of the algorithm will be $O(KC)$. In the experiments, we will show that when choosing the batch size $K = 256$, the algorithm takes about $C = 100,000$ steps to converge. In fact, this algorithm can be computed on a cuda device very quickly. During execution, the policy can be computed in $O(1)$ time.

5.2. Convergence Analysis

Given the above algorithm, in this section, we formally show that the value functions $Q_i, i \in \mathcal{I}$ can converge to an equilibrium point under certain assumptions:

Assumption 1. *Every state $s \in \mathcal{S}$ and action $a_i \in \mathcal{A}$, for $i \in \mathcal{I}$, is visited infinitely often.*

Assumption 2. *The critic learning rates α_t for optimizing Equation (11) satisfy $\sum_{\alpha_t=0}^{\infty} \alpha_t(s, \mathbf{a}) = \infty$, and $\sum_{\alpha_t=0}^{\infty} [\alpha_t(s, \mathbf{a})]^2 < \infty$ holds uniformly with probability 1.*

Assumption 3. *The aggregated message m_g is a representation of the global state information s and action $\mathbf{a}$.*

Assumption 4. *The stage game at each interval t has a global optimal point. The global points are selected by our algorithm to update the critic functions with probability 1.*

Assumptions 1 and 2 are weak ones that are easy to meet. Assumption 3 is met if (1) the communication message m_g can encode the entire state without information loss; (2) every other vehicle's policy can be inferred based on m_g. The two conditions are reasonable according to our communication-based framework. Assumption 4 is a strong assumption. It may not be easily met. However, our empirical experiments demonstrate that this assumption is satisfied mostly since the algorithm can converge in different scenarios. The convergence result mainly originates from the following lemma [40]:

Lemma 1. ***(Szepesvari and Littman (1999), Corollary 5)*** *Assume ρ_t satisfies Assumption 2 and the mapping $P^t : \mathbb{Q} \to \mathbb{Q}$ has the following condition: there exists a number $0 < \gamma < 1$ and a sequence $\lambda_t \geq 0$ converging to zero with probability 1 such that $\left\| P^t Q - P^t Q^* \right\|_\infty \leq \gamma \| Q - Q^* \|_\infty + \lambda_t$ for all $Q \in \mathbb{Q}$ and $Q^* = \mathbb{E}[P^t Q^*]$, then the iteration defined by*

$$Q' = (1 - \rho_t)Q + \rho_t[P^t Q] \tag{15}$$

converges to Q^* with probability 1.

According to Assumption 3, the messages m_g is a compact representation of the global state s and actions $\mathbf{a}$. Therefore, there is $Q_i(s, \mathbf{a}) = Q_i(o_i, a_i, m_g)$. Define the transition function P^t and the convergence point Q^* as

Definition 1. *Let $P^t : \mathbb{Q} \to \mathbb{Q}$ be a mapping on the complete metric space $\mathbb{Q} \to \mathbb{Q}$, $P^t Q = (P^t Q_1, P^t Q_2, \ldots, P^t Q_N)$, where*

$$P^t Q_i(s, \mathbf{a}) = r_i + \gamma Q_i(o_i', \mu_i(o_i', \bar{}^m(s')), \bar{}^m(s'))) \tag{16}$$

for $i \in \mathcal{I}$, where $\bar{}^m(\cdot) = (\mu_1^m, \ldots, \mu_N^m)$.

Definition 2. *Q^* is the convergence point if it satisfies*

$$\begin{aligned} Q_i^*(o_i, o_i, m_g) = r_i + \gamma \sum_{s' \in S} p(s' \mid s, \mathbf{a}) \\ Q_i^*(o_i', \mu_i(o_i', \bar{}^m(s'), \mu^m(s'))) \end{aligned} \tag{17}$$

With the above definitions, we show that the transition function P^t is a "contraction mapping" with the fixed point at Q^*.

Lemma 2. *The convergence point is a fixed point:* $\mathbb{E}[P^t Q^*] = Q^*$.

Proof. Since Q^* is a convergence point in the game, the vehicles will still follow the current policy μ^*. According to the Bellman equation (Equation (1)), there is

$$\begin{aligned} & Q_i^*(o_i, a_i, m_g) \\ & = r_i + \gamma \sum_{s' \in S} p(s' \mid s, \mathbf{a}) Q_i^*(o_i', {}^{-m}(s'), \mu_i(o_i', {}^{-m}(s'))) \\ & = \sum_{s' \in S} p(s' \mid s, \mathbf{a})(\gamma Q_i^*(o_i', {}^{-m}(s'), \mu_i(o_i', {}^{-m}(s'))) + r_i) \\ & = \mathbb{E}[P^t Q_i^*(o_i, x_i, m_g)] \end{aligned} \tag{18}$$

where the forth line takes the expectation from $p(s' \mid s, \mathbf{a})$ and the Bellman equation. □

Next, we show that P^t is a "contraction mapping". According to Assumption 3, there is $\mu_i(o_i, m_g) = \mu_i(s)$. Similar to [41], the max-norm of the mapping operator can be defined as

$$\begin{aligned} \left\| Q - \hat{Q} \right\|_\infty & \equiv \max_i \mid Q_i - \hat{Q}_i \mid \\ & \equiv \max_{i,s} \mid \gamma Q_i(s) - \gamma \hat{Q}_i(s) \mid \\ & \equiv \max_{i,s,\mathbf{a}} \gamma \mid Q_i(s, \mathbf{a}) - \hat{Q}_i(s, \mathbf{a}) \mid \end{aligned}$$

Lemma 3. $\left\| P^t Q - P^t \hat{Q} \right\|_\infty \leq \gamma \| Q - \hat{Q} \| \infty, \ \forall Q, \hat{Q} \in \mathbb{Q}$.

Proof. According to the transition function P^t, there is

$$\begin{aligned} & \left\| P^t Q - P^t \hat{Q} \right\|_\infty \\ & = \max_{i,s} \gamma \mid Q_i(s, \mathbf{a}) - \hat{Q}_i(s, \mathbf{a}) \mid \\ & = \max_{i,s} \gamma \mid \prod_{j=1}^{N} \mu_j^a(s) Q_i(s) - \prod_{j=1}^{N} \hat{\mu}_j^a(s) \hat{Q}_i(s) \mid \\ & \leq \max_{i,s} \gamma \mid \prod_{j=1}^{N} \mu_j^a(s) [Q_i(s) - \hat{Q}_i(s)] \mid \\ & \leq \max_{i,s} \gamma \mid Q_i(s) - \hat{Q}_i(s) \mid \\ & = \gamma \left\| Q_i - \hat{Q}_i \right\|_\infty \end{aligned} \tag{19}$$

The fourth line of equality comes from our Assumption 3 that the message m_g is a compact representation of s. The fifth line of inequality is from Assumption 4 that the vehicles play the best response with respect to the broadcast message m_g. □

Summarizing the above two lemmas, it is proved that P^t is a "contraction mapping" with the fixed point at Q^*. Thus, according to Lemma 1, there is the following.

Theorem 1. *Under Assumption 1-4, the sequence* $(Q_1, \ldots, Q_N)$ *updated by Algorithm 1 converges a fixed value* $Q^* = (Q_1^*, \ldots, Q_N^*)$.

6. Evaluation

In this section, we first introduce the experiment setup, including the description of the environment, the baselines, and the model parameters. Next, we will show the

performance of our algorithm with comparisons with other baselines. In particular, the results validate the importance of the consensual communication framework.

6.1. Experiment Setup

The Environment To validate the effectiveness of our algorithm, we manually construct a mobile sensing simulation environment based on real historical data set. The data set is collected from a road network from Google Map (Google Map: https://www.google.com/maps, accessed on 10 March 2022), which has the traffic volume at the road network across different hours (the data sets generated during the current study are available in the following https://www.dropbox.com/s/42cl68ns2fud5yk/GOOGLETraffic.zip?dl=0, accessed on 10 March 2022). We focus on an area of 10 km $\times$ 10 km square area centered at $(48.16, 16.33)$. In this map, we uniformly sample 40×40 points as the locations of events. For each position, the traffic volumes are extracted as the event intensities. An illustration of the event map at a given time is presented in Figure 4. The dots represent the events happening at different locations. The events have 5 levels of intensities as $0, 1, 2, 3, 4$. We also add random uniform noise $(0, 1)$ to the event intensities for randomness. Dots with darker colors have higher event intensities. In this map, there assumed to be 5 charging stations at locations of $(8, 32)$, $(32, 8)$, $(8, 8)$, $(32, 32)$ and $(20, 20)$.

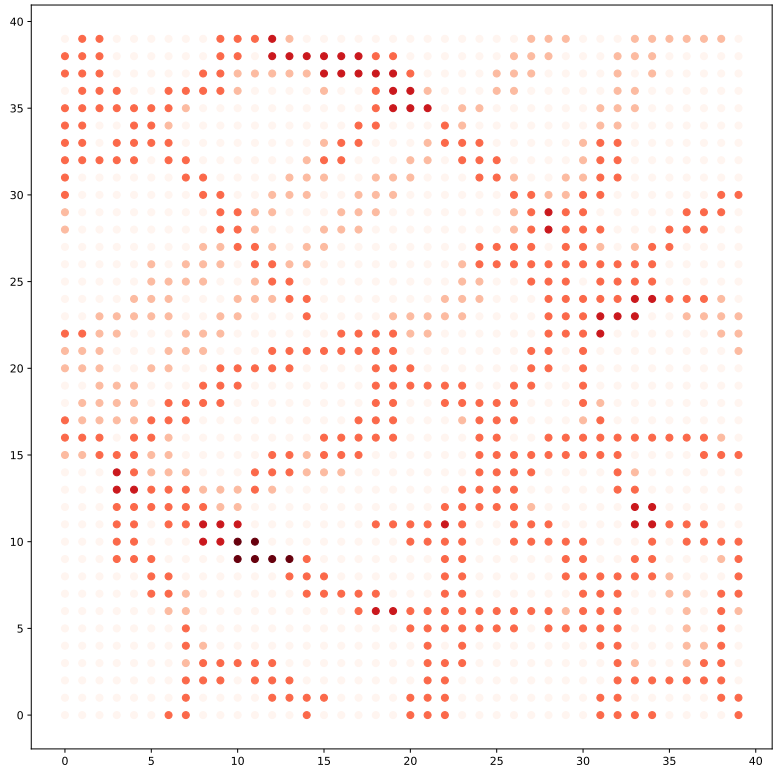

Figure 4. A snapshot of the event intensities in the target.

By default, we suppose the max speed of each vehicle is $S_{\max} = 2$, and the sensing radius is $l_i = 2$. Therefore, each vehicle can cover multiple events at the same time. The battery capacity of each vehicle is $b_{\max} = 40$. During moving, the coefficient of battery consumption is $\beta = 1$, $\Delta_0 = 1$. The vehicles can regularly navigate to the charging station, where they will be recharged $b_0 = 20$ units of battery at each time step. The penalty of running out of power is set as $c = 40$. We will also try other values to validate the effectiveness of our algorithm. A small size of the replay buffer is set as 10^5, since the vehicles policies may be dynamic.

Baselines We name our algorithm as ConComm (CONsensual COMMunication), and compare with the following baselines which can generate continuous actions.

- ConComm (no MI): In this algorithm, we implement the ConComm algorithm without the mutual information item. This comparison is to demonstrate the effectiveness of the mutual information item.
- DDPG [15]: In this algorithm, each mobile vehicle independently learns a policy to schedule the sensing path. The main drawback is that the multi-agent environment does not follow the Markov property, which may lead to the failure of this algorithm.
- MADDPG [9]: MADDPG uses the CTDE framework, where there is a global critic function that has access to the historical samples from all mobile vehicles. However, the policies of the vehicles are not coordinated explicitly during execution.
- MAPPO [42]: This algorithm is a multi-agent version of PPO. It has achieved state-of-the-art performance in many scenarios.

Model Parameters For different algorithms, we use similar critic network structures with an FC layer with 64 hidden units. The FC layer is followed by a ReLU activation layer for non-linear activation. The output is connected with a GRU layer with 64 hidden units and then fed into another FC layer to output the critic value. The actor networks have a similar structure. The only difference is the output of the networks. The communication actor network outputs a message with size 6 followed by a sigmoid layer to restrict the message in the range $(0, 1)$. The messages are aggregated with a MEAN operator, i.e., $m_g = \frac{1}{N}\sum_{i\in\mathcal{I}} m_i$. The moving actor network outputs a vector of size 2, followed by a sigmoid layer to restrict the range of the speed and angle. Maximum speed and angle are used to project the outputs into new ranges. For the variation network, the input is the embedding after the FC layer. It is then fed into two FC layers with 64 hidden units to predict the aggregate message. Mean squared error is used as the loss function for the variation network. The weight of the MI item ρ is set as 0.5 so that different parts of the reward function are comparable.

6.2. Performance Analysis

Convergence of Training In the first experiment, we assume there are $N = 12$ mobile vehicles, and examine the convergence of the algorithms during training in Figure 5. The average step reward is evaluated every 200 steps. We assume different vehicles share the same network parameters. Nonetheless, the vehicles can still behave differently with local observations. The y-axis represents the average step reward for each vehicle $r = \frac{1}{N}\sum_{i\in\mathcal{I}} r_i$. Each of the RL-based algorithms is trained 3 times. The shaded area represents one standard deviation. As presented, our proposed ConComm achieves the highest performance at most of the time. The average step reward of ConComm can converge to around 17 after about only 50,000 steps. The performance then stabilizes around at this level. Moreover, the variance of ConComm is also more stable compared to others. This is because the vehicles are more likely to have coordinated behaviors. ConComm (no MI) is the algorithm without explicit policy coordination. The result can be relatively high due to the communication among the mobile vehicles. However, the performance is worse than ConComm, which validates the effectiveness of the MI item. DDPG has the worst performance among the algorithms. This is mainly due to the fact that the vehicles make decisions independently. Therefore, there may be lots of repeated sensing efforts among the vehicles. MADDPG and MAPPO have similar performances that are slightly better than DDPG. The main reason is that they adopt the "centralized training, decentralized execution" mechanism. However, in the execution phase, there may still be uncoordinated behaviors with unseen environment states. Different vehicles may not achieve consensus before making decisions. The above comparisons show that communication plays an important role in coordinating the vehicles' behaviors.

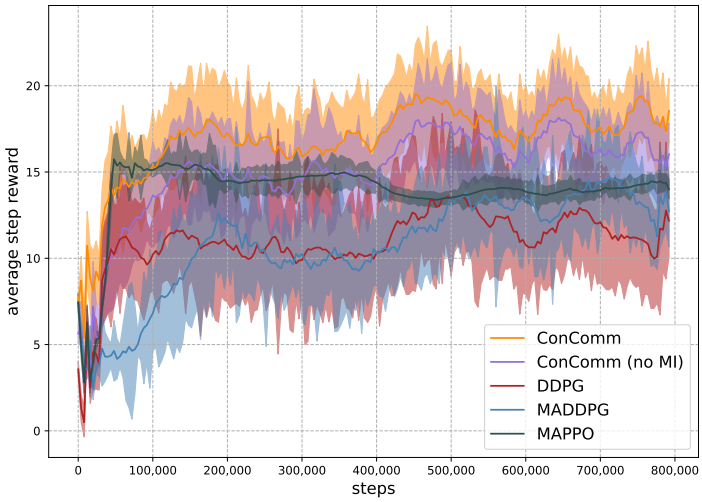

Figure 5. The performance of different algorithms in the execution phase.

Performance during Execution After training, we fix the network parameters and compare the performance of different algorithms in the simulation environment without exploration. The results are shown in Figure 6. In this figure, the height of each bar represents the sensing reward, where the red part is the battery penalty, and the blue part is the true average reward, which equals the sensing reward minus the battery penalty. The algorithm with the highest blue bar has the best performance.

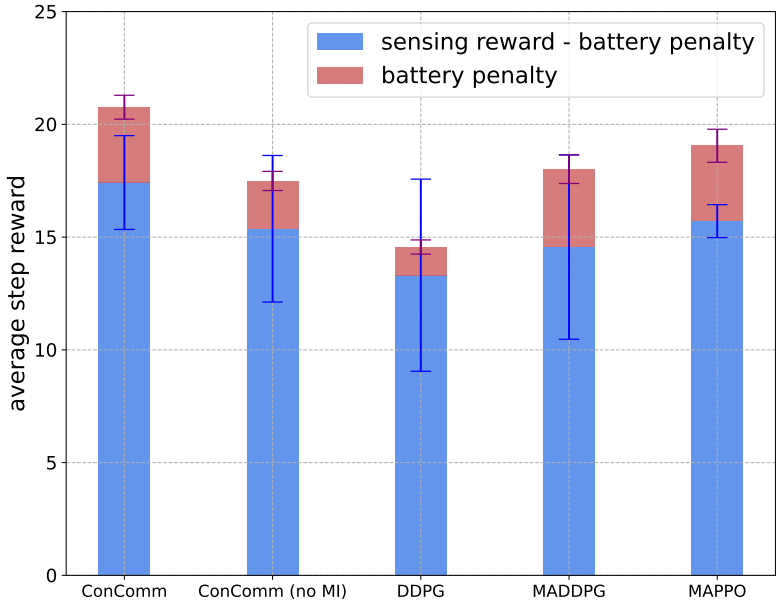

Figure 6. The performance of different algorithms in the execution phase.

As presented, our proposed ConComm achieves the best performance (the blue part) among the algorithms. In particular, the sensing reward (the blue+red part) also outperforms other algorithms significantly. This is because the vehicles in ConComm can avoid repeated sensing through communication. The ConComm (no MI) can also have high performance. It achieves lower battery penalty because the vehicles' behaviors will not be affected by the communication messages explicitly. DDPG also performs well in charging since each vehicle only cares about its own reward. However, the global sensing reward can be quite limited, which may be caused by the lack of coordination. For the

MADDPG and MAPPO algorithms, as they lack the mechanism of coordination in the execution phase, they may not perform as well as our ConComm algorithm. In summary, to achieve high performance, the vehicles should not only try to sense more events with the limited battery, but they need also coordinate with others to avoid repeated sensing.

We also investigate the trajectories of the vehicles in our ConComm to show the effectiveness. We collect the vehicles' trajectory in the execution phase for 1000 steps and obtain the appearance count in the map. The appearance counts are normalized and plotted as a heatmap. The result is presented in Figure 7. In the heatmap, areas with a redder color are visited more often by the mobile vehicles, and the blue areas are visited less often. Compared with Figure 4, the areas where the event intensities are higher also have more vehicle appearances. These areas are dispersed since the vehicles can cooperate to maximize the coverage and reduce repeated sensing. Moreover, the areas near the charging stations also have redder colors; this is because the vehicles regularly moves to the stations for charging. Above all, the heatmap validates that the vehicles of ConComm can not only navigate back for charging, but also properly move to the areas with high event intensities. This heatmap illustrates that our proposed ConComm can properly coordinate the navigation of the vehicles.

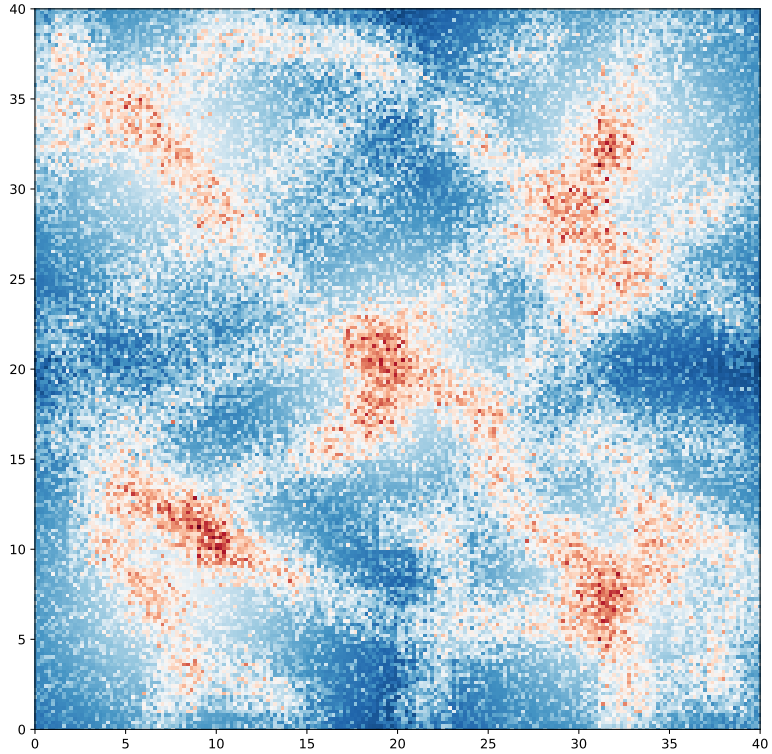

Figure 7. The heatmap of the vehicles trajectories of ConComm.

Policy Coordination via Communication The above two experiments have already shown that explicitly coordinating the policies of different mobile vehicles is crucial for cooperative sensing. In this part, we investigate the effect of coordination by adjusting the weight of the MI item. In addition to the default value $\rho = 0.5$, we change the weight ρ to different values from 0 and 1 and observe the convergence process during training. Note that when ρ is 0, the algorithm degrades to the case of ConComm (no MI). When ρ is 1, the vehicles neglect the sensing reward and battery penalty, and focus only on coordinating with others.

The results are shown in Table 2. As presented, introducing the policy coordination can significantly improve the performance when ρ is non-zero. This validates that positive communication is necessary for coordinating the decentralized vehicles. Meanwhile, when the coefficient is too large, the performance may decrease since the vehicles care more about

coordination and less about sensing reward. When the coefficient reaches 1, the vehicles focus only on the coordination and thus the sensing reward is very poor. The results show that the vehicles need to balance between coordination and sensing. The performance will degrade if focusing on only one of them.

Table 2. Different parts of the performance of ConComm. The average step reward can be described as the difference between the sensing reward and the battery penalty.

ρ	Sensing Reward	Battery Penalty	Average Step Reward
0	17.33	2.14	15.19
0.25	20.83	3.20	17.35
0.5	20.76	3.34	17.42
0.25	19.58	4.20	15.38
1	7.87	37.36	−29.49

Validating the Variation Network In this part, we show that the communication message indeed influences the vehicles' moving policy. More concretely, we compute the cross entropy between the policy μ_i and the neighbors' aggregated message m_g as $\mathcal{H}(\mu_i, m_g)$. The policy μ_i is represented as the historical trajectories $traj_i$. Cross entropy measures the average number of bits needed to identify an event drawn from the set if a coding scheme used for the set is optimized for an estimated probability distribution, rather than the true distribution. It can also be regarded as the distance between the two probability distributions. A low cross entropy distance indicates that the two probability distributions could have high correlation.

We present the dynamics of the cross entropy during training in Figure 8. As presented, the cross entropy is high at the beginning. This is because the vehicles have not learned to correlate with the communication message. As the training proceeds, the cross entropy value becomes lower and stabilizes at about 1.0. This validates that the policy μ_i becomes more correlated with the communication.

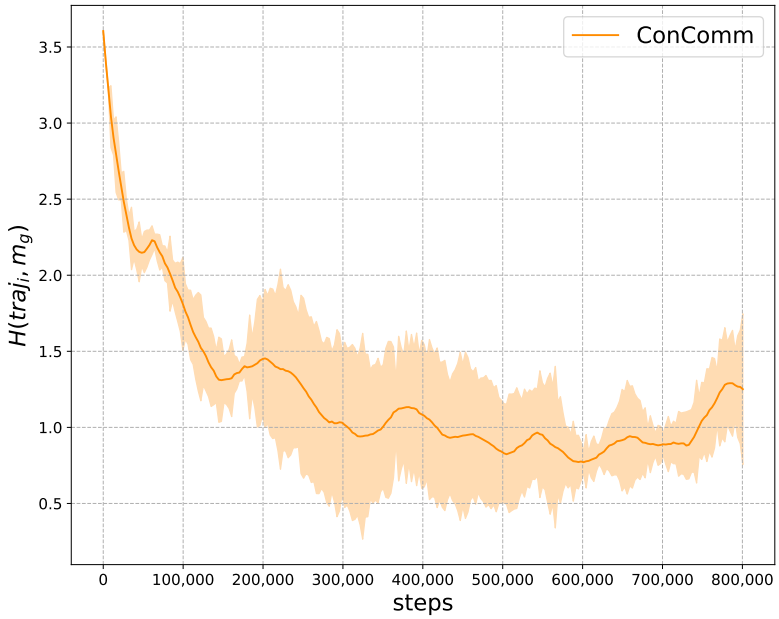

Figure 8. The cross entropy between the vehicles aggregated message m_g and moving policy μ_i represented as $traj_i$. The value is lower when the message is more correlated with the vehicles' moving policies.

Validation the Penalty of Battery Next, we investigate the effect of the hyper-parameter c in shaping the battery penalty. Generally, with a larger value of c, the vehicles will navigate to the charging station more frequently to avoid running out of battery. In practice, this parameter can be set freely by the vehicles and our algorithm can adapt to different values of c. In this experiment, we train the algorithms with different values of c and validate the performance with the default value $c = 40$. Figure 9 presents the results.

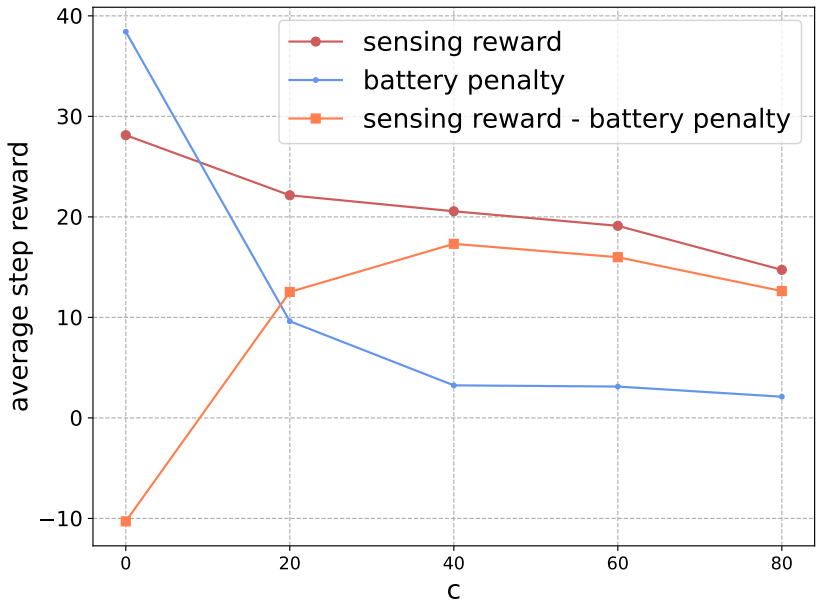

Figure 9. The average step reward (orange part), sensing reward (red part) and battery penalty (blue part) in the execution phase with respect to different values of c.

As presented, when c is 0, the vehicles will not care about the battery penalty and focus only on sensing events. Therefore, there will be high sensing reward, but the battery penalty will also be very high, leading to low average step reward. When the value of c increases, the vehicles will be more conservative to run out of power. They will have low battery penalty. However, the sensing reward will also decrease. In general, choosing a proper value of c can balance the preference of sensing and battery. In practice, we can set the value of c as the cost of reclaiming the vehicles when they run out of power. If this is unacceptable, we can also enforce the vehicles to navigate back to the charging station if needed.

Scalability In this last experiment, we validate the scalability of ConComm. We increase the number of mobile vehicles to 128 and charging stations to 16. The map is divided into 80×80 grid space with charging stations randomly and uniformly distributed. Similar to that above, we assume the mobile agents share the same network parameters. The algorithms of MADDPG and MAPPO would take too much time, so we only present the result of ConComm, ConComm (no MI) and DDPG. As shown in Figure 10, the ConComm algorithm can still achieve better performance. When there are more agents, they may become more easy to coincide. So the average step reward will be lower than before. Nonetheless, ConComm can still successfully coordinate the behaviors of the agents and achieve high performance. In this case, as there is no explicit coordination, the variance of ConComm (no MI) will be larger. The result of DDPG is also not stable since the vehicles' policies are mostly dynamic, leading to low efficiency of coordination. Moreover, the performance of DDPG will even degrade after about 600,000 steps. This may result in the DDPG agents being not coordinated and falling into local optimal points.

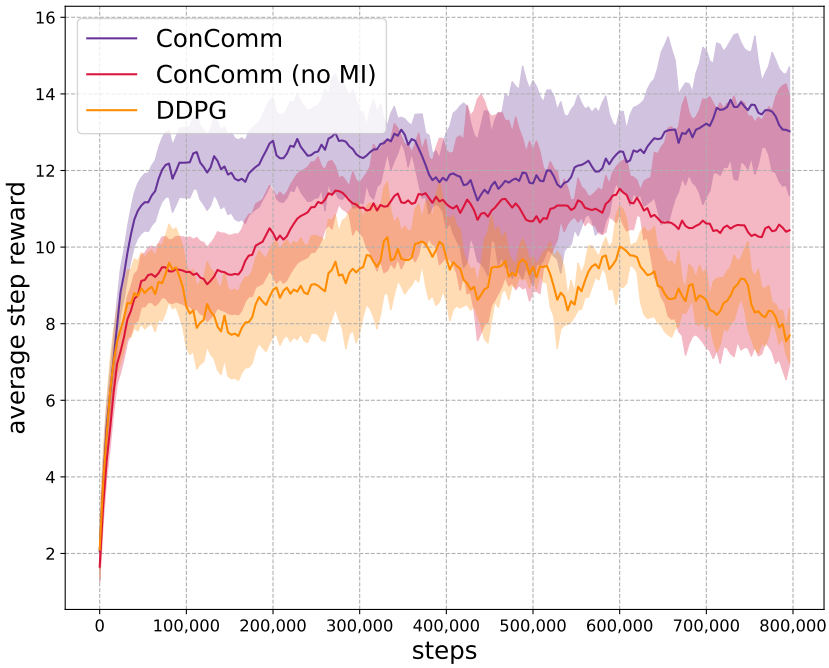

Figure 10. The scalability.

7. Conclusions

This paper studies the problem of mobile sensing in an open, dynamic environment. To maximize the long-term spatial–temporal coverage of the events, we propose a decentralized policy coordination framework. The main idea is to introduce a communication mechanism among the mobile vehicles. On one hand, the vehicles can share local information with each other to break through the dilemma of decentralized execution; on the other hand, the vehicles can have coordinated behavior with enforced positive communication. In particular, the consensual communication is achieved by maximizing the mutual information between the received message and the policy. We conduct extensive experiments to validate the performance of our algorithm. The results show that our algorithm can converge very fast in the training phase, and outperforms other baselines significantly in the execution phase. Moreover, the experiments show that the consensual communication mechanism plays an important role in coordinating the behaviors.

For future works, we aim to extend the current method from two aspects. First, the battery constraints in this paper are relaxed as part of the objective, and may lead to violations. Therefore, we need to devise method with "hard" constraints. Second, we will improve the interpretability of the communication messages to understand the internal mechanism that promotes the cooperation among the vehicles.

Author Contributions: B.Z.: conceptualization, theoretical analysis, formal analysis, writing—original draft preparation; L.W.: methodology and supervision; I.Y.: validation, review, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by National Key Research and Development Program of China under Grant No. 2019YFB2101704; National Natural Science Foundation of China under Grant No. 62202238; Natural Science Foundation of Jiangsu Province under Grant No. BK20200752.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The traffic data address: https://www.dropbox.com/s/42cl68ns2fud5yk/GOOGLETraffic.zip?dl=0 (accessed on 23 October 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gao, Y.; Dong, W.; Guo, K.; Liu, X.; Chen, Y.; Liu, X.; Bu, J.; Chen, C. Mosaic: A low-cost mobile sensing system for urban air quality monitoring. In Proceedings of the 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
2. Carnelli, P.; Yeh, J.; Sooriyabandara, M.; Khan, A. Parkus: A Novel Vehicle Parking Detection System. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA 4–9 February 2017, Volume 31, pp. 4650–4656.
3. Laport, F.; Serrano, E.; Bajo, J. A Multi-Agent Architecture for Mobile Sensing Systems. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 4439–4451. [CrossRef]
4. Ranieri, A.; Caputo, D.; Verderame, L.; Merlo, A.; Caviglione, L. Deep Adversarial Learning on Google Home Devices. *J. Internet Serv. Inf. Secur.* **2021**, *11*, 33–43.
5. Liu, C.H.; Ma, X.; Gao, X.; Tang, J. Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning. *IEEE Trans. Mob. Comput.* **2019**, *19*, 1274–1285. [CrossRef]
6. Wei, Y.; Zheng, R. Multi-Robot Path Planning for Mobile Sensing through Deep Reinforcement Learning. In Proceedings of the IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
7. Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; Whiteson, S. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning, In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4292–4301.
8. Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual Multi-Agent Policy Gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA 2–7 February 2018; Volume 32.
9. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, O.P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 6379–6390.
10. Foerster, J.; Assael, I.A.; de Freitas, N.; Whiteson, S. Learning to Communicate with Deep Multi Agent Reinforcement Learning. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Volume 29, pp. 2137–2145.
11. Cao, K.; Lazaridou, A.; Lanctot, M.; Leibo, J.Z.; Tuyls, K.; Clark, S. Emergent Communication through Negotiation. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
12. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
13. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
14. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 387–395. .
15. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
16. Lowe, R.; Foerster, J.; Boureau, Y.L.; Pineau, J.; Dauphin, Y. On the pitfalls of measuring emergent communication. *arXiv* **2019**, arXiv:1903.05168.
17. Sukhbaatar, S.; Szlam, A.; Fergus, R. Learning multiagent communication with backpropagation. *arXiv* **2016**, arXiv:1605.07736.
18. Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M.; Pineau, J. Tarmac: Targeted multi-agent communication. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 1538–1546.
19. Jaques, N.; Lazaridou, A.; Hughes, E.; Gulcehre, C.; Ortega, P.A.; Strouse, D.; Leibo, J.Z.; de Freitas, N. Intrinsic social motivation via causal influence in multi-agent RL. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
20. Karaliopoulos, M.; Telelis, O.; Koutsopoulos, I. User recruitment for mobile crowdsensing over opportunistic networks. In Proceedings of the 2015 IEEE Conference on Computer Communications, Hong Kong, China, 26 April–1 May 2015; pp. 2254–2262.
21. Hu, Q.; Wang, S.; Cheng, X.; Zhang, J.; Lv, W. Cost-efficient mobile crowdsensing with spatial-temporal awareness. *IEEE Trans. Mob. Comput.* **2019**, *20*, 928–938. [CrossRef]
22. Rahili, S.; Lu, J.; Ren, W.; Al-Saggaf, U.M. Distributed coverage control of mobile sensor networks in unknown environment using game theory: Algorithms and experiments. *IEEE Trans. Mob. Comput.* **2017**, *17*, 1303–1313. [CrossRef]
23. Esch, R.R.; Protti, F.; Barbosa, V.C. Adaptive event sensing in networks of autonomous mobile agents. *J. Netw. Comput. Appl.* **2016**, *71*, 118–129. [CrossRef]
24. Li, F.; Wang, Y.; Gao, Y.; Tong, X.; Jiang, N.; Cai, Z. Three-Party Evolutionary Game Model of Stakeholders in Mobile Crowdsourcing. *IEEE Trans. Comput. Soc. Syst.* **2021**, *9*, 974–985. [CrossRef]
25. Zhang, C.; Zhao, M.; Zhu, L.; Wu, T.; Liu, X. Enabling Efficient and Strong Privacy-Preserving Truth Discovery in Mobile Crowdsensing. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3569–3581. [CrossRef]
26. Zhao, B.; Liu, X.; Chen, W.N.; Deng, R. CrowdFL: Privacy-Preserving Mobile Crowdsensing System via Federated Learning. *IEEE Trans. Mob. Comput.* **2022**, 1. [CrossRef]

27. You, X.; Liu, X.; Jiang, N.; Cai, J.; Ying, Z. Reschedule Gradients: Temporal Non-IID Resilient Federated Learning. *IEEE Internet Things J.* **2022**, 1. [CrossRef]
28. Nasiraee, H.; Ashouri-Talouki, M.; Liu, X. Optimal Black-Box Traceability in Decentralized Attribute-Based Encryption. *IEEE Trans. Cloud Comput.* **2022**, 1–14. [CrossRef]
29. Wang, J.; Li, P.; Huang, W.; Chen, Z.; Nie, L. Task Priority Aware Incentive Mechanism with Reward Privacy-Preservation in Mobile Crowdsensing. In Proceedings of the 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design, Hangzhou, China, 4–6 May 2022; pp. 998–1003.
30. Komisarek, M.; Pawlicki, M.; Kozik, R.; Choras, M. Machine Learning Based Approach to Anomaly and Cyberattack Detection in Streamed Network Traffic Data. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl.* **2021**, *12*, 3–19.
31. Nowakowski, P.; Zórawski, P.; Cabaj, K.; Mazurczyk, W. Detecting Network Covert Channels using Machine Learning, Data Mining and Hierarchical Organisation of Frequent Sets. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl.* **2021**, *12*, 20–43.
32. Johnson, C.; Khadka, B.; Ruiz, E.; Halladay, J.; Doleck, T.; Basnet, R.B. Application of deep learning on the characterization of tor traffic using time based features. *J. Internet Serv. Inf. Secur.* **2021**, *11*, 44–63.
33. Bithas, P.S.; Michailidis, E.T.; Nomikos, N.; Vouyioukas, D.; Kanatas, A.G. A survey on machine-learning techniques for UAV-based communications. *Sensors* **2019**, *19*, 5170. [CrossRef]
34. An, N.; Wang, R.; Luan, Z.; Qian, D.; Cai, J.; Zhang, H. Adaptive assignment for quality-aware mobile sensing network with strategic users. In Proceedings of the 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, New York, NY, USA, 24–26 August 2015; pp. 541–546.
35. Zhang, W.; Song, K.; Rong, X.; Li, Y. Coarse-to-fine uav target tracking with deep reinforcement learning. *IEEE Trans. Autom. Sci. Eng.* **2018**, *16*, 1522–1530. [CrossRef]
36. Liu, C.H.; Dai, Z.; Zhao, Y.; Crowcroft, J.; Wu, D.; Leung, K.K. Distributed and energy-efficient mobile crowdsensing with charging stations by deep reinforcement learning. *IEEE Trans. Mob. Comput.* **2019**, *20*, 130–146. [CrossRef]
37. Liu, C.H.; Chen, Z.; Zhan, Y. Energy-efficient distributed mobile crowd sensing: A deep learning approach. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1262–1276. [CrossRef]
38. Zeng, F.; Hu, Z.; Xiao, Z.; Jiang, H.; Zhou, S.; Liu, W.; Liu, D. Resource allocation and trajectory optimization for QoE provisioning in energy-efficient UAV-enabled wireless networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 7634–7647. [CrossRef]
39. Samir, M.; Ebrahimi, D.; Assi, C.; Sharafeddine, S.; Ghrayeb, A. Leveraging UAVs for coverage in cell-free vehicular networks: A deep reinforcement learning approach. *IEEE Trans. Mob. Comput.* **2020**, *20*, 2835–2847. [CrossRef]
40. Szepesvári, C.; Littman, M.L. A unified analysis of value-function-based reinforcement-learning algorithms. *Neural Comput.* **1999**, *11*, 2017–2060. [CrossRef]
41. Hu, J.; Wellman, M.P. Nash Q-learning for general-sum stochastic games. *J. Mach. Learn. Res.* **2003**, *4*, 1039–1069.
42. Yu, C.; Velu, A.; Vinitsky, E.; Wang, Y.; Bayen, A.; Wu, Y. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. *arXiv* **2021**, arXiv:2103.01955.