Joint Charging and Relocation Recommendation for E-Taxi Drivers via Multi-Agent Mean Field Hierarchical Reinforcement Learning

Enshu Wang, Rong Ding, Zhaoxing Yang, Haiming Jin, *Member, IEEE*, Chenglin Miao, *Member, IEEE*, Lu Su, *Member, IEEE*, Fan Zhang, *Member, IEEE*, Chunming Qiao, *Fellow, IEEE*, and Xinbing Wang, *Senior Member, IEEE*

Abstract—Nowadays, most of the taxi drivers have become users of the relocation recommendation service offered by online ride-hailing platforms (e.g., Uber and Didi Chuxing), which could oftentimes lead drivers to places with profitable orders. At the same time, electric taxis (e-taxis) are increasingly adopted and gradually replacing gasoline taxis in today's public transportation systems due to their environmental-friendly nature. Though effective for traditional gasoline taxis, existing relocation recommendation schemes are rather suboptimal for e-taxi drivers' user experience. On one hand, the existing schemes take no account of taxis' refueling decisions, as the refueling durations of gasoline taxis are usually short enough to be ignored. However, the charging duration of the e-taxis spent at charging stations can be as long as hours. Obviously, an e-taxi's battery could be easily depleted by the continuous relocations suggested by existing schemes, and thus will have to be charged for a long time afterwards, making the e-taxi driver miss numerous order-serving opportunities. On the other hand, charging posts are typically sparsely and unevenly distributed across a city. With no consideration of charging opportunities, existing schemes could probably send an e-taxi to an area with no charging post around, even though its battery is running low. To optimize e-taxi drivers' user experience, in this paper, we design a joint *charging and relocation recommendation system for e-taxi drivers (CARE)*. We take the perspective of e-taxi drivers and formulate their decision making as a multi-agent reinforcement learning problem where each e-taxi driver aims to maximize his own cumulative rewards. More specifically, we propose a novel *multi-agent mean field hierarchical reinforcement learning (MFHRL)* framework. The hierarchical architecture of MFHRL helps the proposed CARE provide far-sighted charging and relocation recommendations for e-taxi drivers. Besides, we integrate each hierarchical level of MFHRL separately with the mean field approximation to incorporate e-taxis' mutual influences in decision making. We set up a simulator with one of the largest real-world e-taxi datasets in Shenzhen, China, which contains the GPS trajectory data and transaction data of 3848 e-taxis from June 1st to June 30th, 2017, coupled with 165 charging stations including 317 fast charging posts and 1421 slow charging posts. We adopt this simulator to generate 6 dynamic urban environments, which reflect the different real-world scenarios faced by e-taxi drivers. In all of these environments, we conduct extensive experiments to validate that the proposed MFHRL framework greatly outperforms all baselines by significantly increasing the rewards obtained by e-taxi drivers. Besides, we also show that the charging policy learned by MFHRL can effectively reduce the range anxiety of e-taxi drivers, which significantly boosts e-taxi drivers' quality of experience.

Index Terms—Electric taxi, hierarchical reinforcement learning, multi-agent reinforcement learning, mean field approximation

1 Introduction

Recently, the way taxi drivers work has been fundamentally revolutionized by online ride-hailing platforms such as Uber[1] and Didi Chuxing[2]. Instead of simply relying on their own experience to search for potential passengers, taxi drivers nowadays tend to follow the relocation recommendations given by such platforms, which could oftentimes lead them to places with profitable orders. At the same time, taxi drivers are also experiencing a rapid vehicle electrification process [1], [2], [3], [4] due to the environment-friendly nature of electric vehicles. To advocate green commuting, several countries such as the United States and China have extensively adopted electric taxis (e-taxis) to replace gasoline taxis.

- *Enshu Wang, Lu Su, and Chunming Qiao are with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260 USA. E-mail: {enshuwan, lusu, qiao}@buffalo.edu.*
- *Rong Ding and Zhaoxing Yang are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {dingrong, yiannis}@sjtu.edu.cn.*
- *Haiming Jin is with the John Hopcroft Center for Computer Science and the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: jinhaiming@sjtu.edu.cn.*
- *Chenglin Miao is with the Department of Computer Science, University of Georgia, Athens, GA 30602 USA. E-mail: cmiao@uga.edu.*
- *Fan Zhang is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China. E-mail: zhangfan@siat.ac.cn.*
- *Xinbing Wang is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: xwang8@sjtu.edu.cn.*

Manuscript received 12 June 2020; revised 25 Aug. 2020; accepted 28 Aug. 2020. Date of publication 7 Sept. 2020; date of current version 4 Mar. 2022.
(Corresponding author: Haiming Jin.)
Digital Object Identifier no. 10.1109/TMC.2020.3022173

1. https://www.uber.com/
2. https://www.didiglobal.com/

Though making reasonably good suggestions for gasoline taxis, existing relocation recommendation schemes are far from optimal for the e-taxis. On one hand, existing schemes take no account of taxis' refueling decisions, as gasoline taxis can usually be refueled in minutes short enough to be ignored. However, unlike gasoline taxis, the charging time of the e-taxis spent at charging stations can be as long as hours. The continuous relocations suggested by existing schemes will easily deplete an e-taxi's battery. As a result, an e-taxi driver has to charge his vehicle for a long time afterwards, who will inevitably miss the following numerous opportunities to serve orders. On the other hand, in practice, charging posts are typically sparsely and unevenly distributed across a city. Without considering charging opportunities, existing schemes could probably relocate an e-taxi to an area with no charging post around, even though its battery is running low. Therefore, the charging and relocation decisions have to be jointly scheduled for the benefit of e-taxi drivers.

In this paper, we design a *charging and relocation recommendation system for e-taxi drivers (CARE)*[3] to jointly schedule e-taxis' charging and relocation decisions. Making such recommendations for an e-taxi driver is naturally a sequential decision-making task that aims to maximize his long-term cumulative reward (e.g., the number of orders he serves over a long period), which is challenging in the following aspects. First, e-taxi drivers work in a real-world urban environment with complex dynamics (e.g., stochastic order arrivals), which requires them to take adaptive actions as the environment evolves. Second, actions such as charging or traveling to distant locations for potential orders are beneficial to e-taxi drivers in the long run but provide no immediate reward. Clearly, without immediate incentives, it is difficult for a recommendation system to make such far-sighted decisions. Last but not least, e-taxi drivers are non-cooperative with each other and may compete for the same charging or order-serving opportunities. Consequently, an e-taxi driver's actions will be affected by the others, and thus integrating such influences in making charging and relocation recommendations becomes essential. However, given that there could be as many as thousands of e-taxis in a city-scale area at the same time, it is then extremely hard to capture the aggregate influences of the other e-taxis on any specific one.

To address these challenges, we propose a novel multi-agent mean field hierarchical reinforcement learning (MFHRL) framework, which integrates each level of hierarchical reinforcement learning [5] with the mean field approximation [6]. The proposed MFHRL framework treats each e-taxi driver as an agent and generates adaptive charging and relocation actions for each e-taxi driver in the dynamic urban environment.

In particular, inspired by [5], we employ a two-level hierarchical architecture in MFHRL, where each agent is composed of a *manager* module and a *worker* module. Between them, the manager operates at a lower temporal resolution but can look ahead for a long time interval. More specifically, the manager sets goals which intrinsically guide the worker's decisions, as the worker will be rewarded for following the goals. In contrast, the worker focuses on a short interval with a higher temporal resolution. It outputs the agent's decisions by jointly considering the goals set by the manager and the rewards obtained from the environment. In this way, though, without any immediate reward from the environment, the worker is still willing to make the far-sighted charging and relocation decisions, because of the intrinsic rewards from the manager when those actions conform with the goals. As a result, such hierarchical architecture enables an e-taxi to take far-sighted actions that optimize the long-term reward.

To further incorporate e-taxis' mutual influences in decision making, we integrate the aforementioned hierarchical architecture with the mean field approximation [6], which approximates the influences of thousands of e-taxis in the city-scale area by the *average* effect from a limited number of e-taxis within a local area. In our design, we integrate each of the two hierarchical levels separately with the mean field approximation. Specifically, since the worker focuses on a short time interval, the mean field approximation in the worker module only considers the influences of the e-taxis within its near neighborhood. In contrast, to capture the possible influences of other agents' far-sighted actions, the mean field approximation in the manager module considers the influences of the e-taxis in a broader area reachable by the e-taxi within the time interval covered by the manager.

To summarize, the main contributions of this paper are listed as follows.

- For the first time, we design a *charging and relocation recommendation system for e-taxi drivers (CARE)* to jointly schedule charging and relocation decisions for thousands of e-taxis in a city-scale area. Based on this idea, we take the perspective of e-taxi drivers and formulate this problem as a multi-agent reinforcement learning problem, where each e-taxi driver aims to maximize his own long-term reward.
- To solve this problem, we propose a novel *multi-agent mean field hierarchical reinforcement learning (MFHRL)* framework. The hierarchical architecture of MFHRL can set goals for the agents to effectively learn far-sighted charging and relocation decisions. Besides, we integrate each of the two hierarchical levels separately with the mean field approximation to incorporate agents' mutual influences in decision making. To the best of our knowledge, MFHRL is the first multi-agent reinforcement learning framework that integrates hierarchical reinforcement learning with mean field approximation.
- Our study is based on one of the largest real-world e-taxi datasets, which consists of (i) 3848 e-taxis, (ii) around 168000 orders per day, (iii) 164 charging stations which involve 317 fast charging posts and 1421 slow charging posts. Based on the dataset, we set up an e-taxi simulator and conduct extensive experiments. The results show that the proposed MFHRL framework outperforms all the baselines.

2 Related Work

In this section, we first discuss the related work on the electric vehicle (EV) and then describe the related work on deep

3. The name CARE comes from Charging and relocAtion REcommendation.

reinforcement learning. In what follows, we will show the novelty of this paper compared with past literature both in terms of the investigated problem and the solution techniques.

2.1 Research on Electric Vehicle

Nowadays, EVs are becoming increasingly more popular. Studies on how to improve the experience of EV drivers in urban transportation systems [1], [2], [3], [4], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21] have drawn significant attention. In this subsection, we organize the related work on this topic into two categories, i.e., charging station deployment and charging station recommendation.

2.1.1 Research on Charging Station Deployment

The objective of studies [1], [3], [14], [15], [16], [17] on charging station deployment is mainly to improve the charging and operational efficiency of EV drivers. More specifically, Yan *et al.* [3] propose a multi-objective charging station deployment scheme. Such a scheme aims to support the continuous operation of e-taxis and meanwhile maximize the e-taxis' opportunities of picking up passengers at the charging stations. Li *et al.* [16] propose an optimal charging post placement scheme for EV drivers to minimize the average driving time to the nearest available charging posts. Sarker *et al.* [1] propose a charging station deployment scheme that minimizes the drivers' anxiety about the limited driving range of EVs and the availability of charging posts. Based on the proposed deployment scheme, the authors design a multi-object route planning system that can always provide an energy-efficient route for EV drivers to reduce their anxiety. However, building charging stations and charging posts will bring extra costs. Apart from the cost of the land resources which are already very scarce in large cities, such as New York and Shenzhen, a fast charging station with only 10 charging posts could cost over 358000 dollars[4]. Instead of building new charging posts, in this paper, our proposed recommendation system CARE can effectively utilize the existing limited charging posts, and jointly schedule the charging and relocation actions of numerous e-taxi drivers in a city-scale area to optimize their long-term experience.

2.1.2 Research on Charging Station Recommendation

Recently, EVs such as electric buses (e-buses) and e-taxis have been gradually introduced into the public transportation system, and received wide attention [2], [4], [7], [8], [9], [10], [11], [12], [13], [20]. Wang *et al.* [8] propose a charging station recommendation scheme for e-buses, which recommends charging stations to e-buses in a sequential manner. Different from such a single-agent setting that ignores the mutual influences among e-buses, we study a multi-agent setting and employ the mean field approximation to incorporate mutual influences among thousands of e-taxis in decision making. Moreover, one line of works study the charging station recommendation for e-taxis [2], [4], [20]. Among them, Dong *et al.* [20] propose a real-time charging scheduling system to recommend a charging station with a tightly bounded waiting time. Besides, Wang *et al.* [4] propose a fairness-aware charging station recommendation system to minimize e-taxis' traveling time to charging stations and waiting time at charging stations under the fairness constraint. Furthermore, Wang *et al.* [2] design a real-time charging scheduling system based on the assumption that e-taxis can be charged at e-buses' charging stations. Under such an assumption, the proposed system can recommend a charging station with the minimum traveling and waiting time. However, the above recommendation systems only focus on recommending the charging station locations for e-taxi drivers, who still have to rely on their own experience to decide the charging duration. Different from them, our CARE system can provide drivers the recommendations on when, where, and how long to charge their e-taxis. Apart from charging recommendations, CARE can also provide relocation recommendations that could lead e-taxi drivers to the place with profitable orders. Therefore, as long as the e-taxi drivers follow the charging and relocation recommendations made by CARE, they will have a satisfactory experience.

2.2 Research on Deep Reinforcement Learning

Recently, deep reinforcement learning [22] has been widely adopted to solve sequential decision-making problems in real-world urban environments. Ji *et al.* [23] propose a dynamic ambulance redeployment system via deep reinforcement learning. Besides, deep reinforcement learning is widely used to solve various problems in online ride-hailing platforms [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35] in recent years. Lin *et al.* [26] propose a contextual multi-agent reinforcement learning to maximize the ORR of the ride-hailing platform. Li *et al.* [28] design a deep value-network based reinforcement learning framework to maximize the GMV of the ride-hailing platform. In contrast, we take the perspective of e-taxi drivers and aim to maximize the long-term rewards of each individual e-taxi driver. Besides, these above relocation recommendation schemes [26], [27], [28], [29] are mainly designed for gasoline taxis, which can usually be refueled in minutes that are short enough to be ignored in decision making. However, the charging duration of e-taxis spent at charging stations can be as long as hours. Even worse, e-taxi drivers will receive no immediate reward during this long charging time. Such a phenomenon corresponds to the sparse-reward challenge in reinforcement learning [5], [18], which hinders the reinforcement learning approaches in [26], [27], [28], [29] from learning effective policies in our scenario. However, the hierarchical architecture of our MFHRL framework enables the agents to effectively learn far-sighted charging and relocation decisions. To the best of our knowledge, MFHRL is the first multi-agent reinforcement learning framework integrating hierarchical reinforcement learning with mean field approximation, which outputs far-sighted actions with agents' mutual influences properly considered.

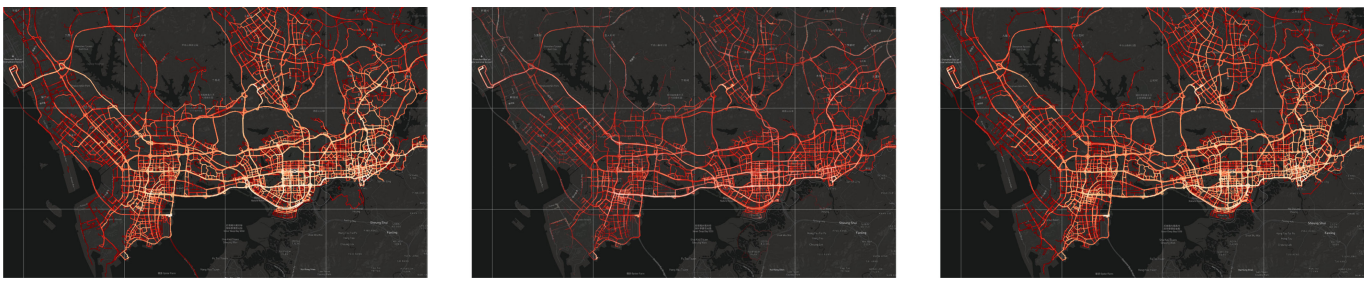
3 Motivation

In this section, we first describe the real-world e-taxi datasets used in this paper. Then, we conduct a comprehensive

4. http://www.council.nyc.gov/data/wp-content/uploads/sites/73/2020/01/Taxi-Medallion-Task-Force-Report-Final.pdf

TABLE 1
An Example of One Piece of Data From Each of Our Datasets

GPS Trajectory	plateID	longitude	latitude	time	status
	BB3296	113.611870	22.616850	2017-06-01 08:11:27	1
Transactions	plateID	pickup time	dropoff time	price (CNY)	distance (m)
	BN1950	2017-06-01 21:48:59	2017-06-01 22:09:21	35.0	4421
Charging Station	stationID	longitude	latitude	# of fast	# of slow
	26	114.131409	22.648233	8	12



(a) Morning peak (6:00-9:00). (b) Noon off-peak (12:00-15:00). (c) Evening peak (18:00-21:00).

Fig. 1. Order distributions in Shenzhen during three different time periods.

data-driven analysis to demonstrate the charging problems that e-taxi drivers are facing.

3.1 Data Description

The real-world datasets used in this paper include the data of e-taxis' GPS trajectories, transactions, and charging stations in Shenzhen from June 1st to June 30th, 2017. An example of one piece of data from each of these datasets is shown in Table 1.

More specifically, a piece of data from the GPS trajectory dataset records the plate ID of an e-taxi, time, longitude and latitude, as well as the corresponding binary status to indicate whether the e-taxi is serving for an order. Given such binary status and the transaction data, we extract the order information that includes the order time, order price, GPS locations of the order origin and destination, as well as order duration with the corresponding pick-up time and drop-off time. The spatial order distributions during 3 different time periods in Shenzhen are shown in Fig. 1, where a brighter color indicates that there are more orders in that area.

The charging station dataset contains the GPS locations of charging stations and the number of fast and slow charging posts they contain. For details, there are totally 164 charging stations which involve 317 fast charging posts and 1421 slow charging posts. All these charging stations and the corresponding charging posts have been built up no later than 2017.

3.2 Charging Problems

Based on the data of e-taxis' GPS trajectories and charging stations, we extract charging activities by using the method in [20] and provide the following data-driven analysis. More specifically, in order to extract the charging activities, we set a circular range with the GPS location of a charging station as the center and 500 meters as the radius. Given the e-taxis' trajectory data, if a sequence of an e-taxi's trajectory points is within this range for a sufficiently long time (e.g., 20 minutes), and meanwhile most of these trajectory points are concentrated around the same GPS location, we regard that this e-taxi driver charges his e-taxi at this charging station.

3.2.1 *Long Charging Duration*

As shown in Fig. 2, there exist 20 percent of the e-taxi drivers who have to spend more than 158 minutes in one day to charge their e-taxis. Such a long charging duration of an e-taxi in one day is almost 20 times longer than the daily refueling time of a gasoline taxi [36], which inevitably results in the missing of numerous order-serving opportunities. Worse still, as shown in Fig. 3, apart from the driving time to charging stations, we can observe that 80 percent of the charging activities cost between 20 to 65 minutes. Thus, without a meticulously designed charging recommendation system, e-taxi drivers have to rely on their personal experience to appropriately schedule their charging activities on when, where, and how long to charge their e-taxis so as to reduce the loss of income.

3.2.2 *Sparse and Uneven Distribution of Charging Stations*

After delivering passengers to their destinations, an e-taxi driver could make a decision on whether to charge his e-taxi or not. Based on our datasets, we calculate the straight line distance from each e-taxi's location to the nearest charging station at the time when the e-taxi driver finishes serving an on-going order. As shown in Fig. 4, we can observe that there exist 20 percent of the e-taxi drivers who have to drive more than 2.5 kilometers to the nearest charging station in the urban area of Shenzhen. Even worse, in the suburban area of Shenzhen, there are 20 percent of the e-taxi drivers who have to drive more than 6 kilometers to the nearest charging station. Besides, it is also worth noting that the result shown in Fig. 4 is about the straight line distance to the nearest

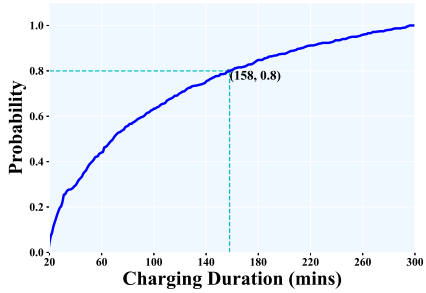


Fig. 2. CDF of the total charging duration of an e-taxi in one day.

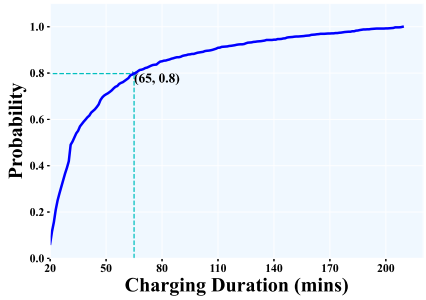


Fig. 3. CDF of the charging duration of an e-taxi for one time of charging.

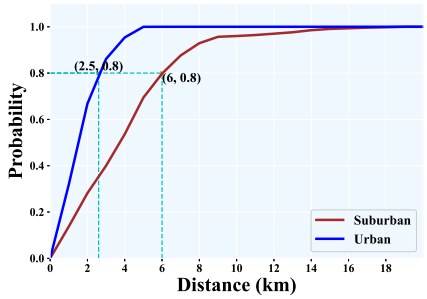


Fig. 4. CDF of the distance between an e-taxi and the nearest charging station when it finishes serving an order.

charging station, which is a lower bound of the actual driving distance. In fact, e-taxi drivers have to drive for a much longer distance when heading to a charging station. Therefore, our observations from Fig. 4 indicates that charging stations are sparsely and unevenly distributed in practice.

3.2.3 Heavy Congestion at Charging Stations

Though there exist 164 charging stations in Shenzhen, we can observe that 80 percent of the charging activities happen in only 37 charging stations, as shown in Fig. 5. Unfortunately, these 37 charging stations are not equipped with sufficient charging posts. As shown in Figs. 6 and 7, the number of the charging activities in both of two different time periods (i.e., 8:00-10:00 and 19:00-21:00) is much larger than that of the charging posts at these charging stations. As a result, there must be a considerable number of e-taxi drivers waiting for charging and suffering from the heavy congestion at the charging stations.

3.2.4 Summary of Data-Driven Analysis

From our data-driven analysis, we find out that e-taxi drivers are suffering from the long charging duration and heavy congestion at charging stations, as well as sparse and uneven

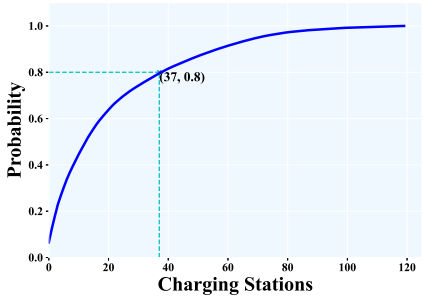


Fig. 5. CDF of the number of charging stations where charging activities concentrate on.

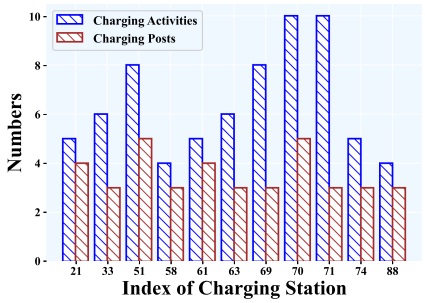


Fig. 6. Number of charging activities at the charging stations during 8:00-10:00.

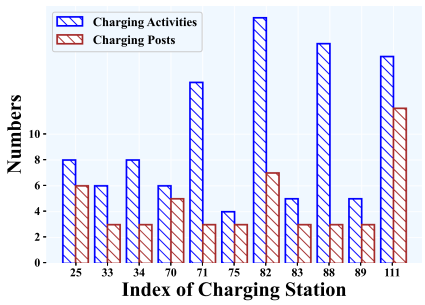


Fig. 7. Number of charging activities at the charging stations during 19:00-21:00.

distribution of charging stations. These inefficient charging problems directly harm the experience of e-taxi drivers and indirectly reduce the time that can be spent in serving orders. With the existence of the above charging problems, it is of vital importance to help e-taxi drivers make proper charging decisions so as to ensure the satisfactory experience for them. Therefore, in this paper, we integrate charging decisions into relocation recommendations and design a recommendation system that jointly schedules charging and relocation decisions for the benefit of e-taxi drivers.

4 Problem Description

We consider a fleet of N e-taxis, denoted as $\mathcal{I} = \{1, \ldots, N\}$, distributed in a city-scale area. In our model, the time horizon is discretized into equal-length time steps, and the entire city is divided into equal-size square grids.

In this paper, we take the perspective of e-taxi drivers and aim to design a *charging and relocation recommendation system for e-taxi drivers (CARE)*. CARE will instruct each e-taxi in every time step to travel to reasonable locations to either serve orders or charge their batteries, as shown in

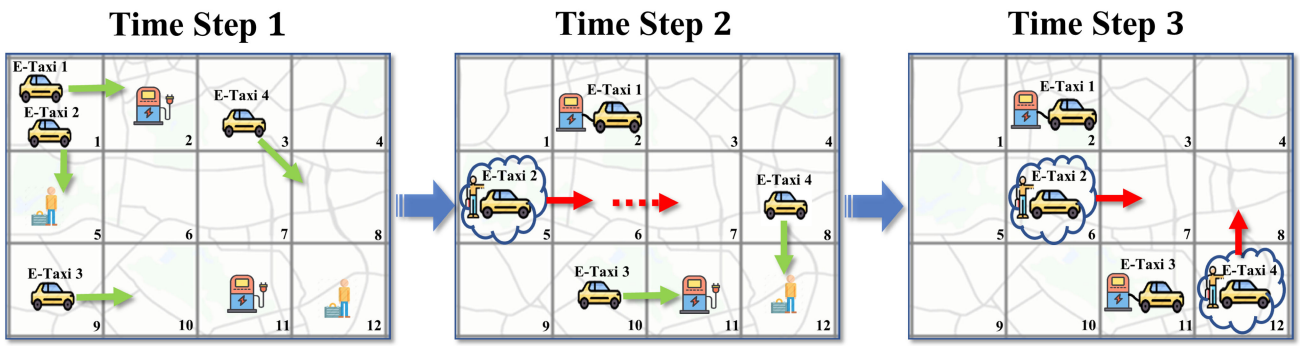


Fig. 8. An example of the recommendations made by CARE over 3 time steps, where the number at the bottom right corner of each grid represents its index. In this example, at time step 1, CARE recommends e-taxi 1 to charge in grid 2, e-taxi 2 to serve an order in grid 5, e-taxi 3 to move to grid 10, and e-taxi 4 to move to grid 8. At time step 2, CARE recommends e-taxi 3 to charge in grid 11, and e-taxi 4 to serve an order in grid 12. E-taxi 2 and 4 take a passenger at time step 2 and 3 respectively, and start traveling to the corresponding destinations.

Fig. 8. Making such recommendations for an e-taxi driver could be treated as a *sequential decision making* problem that aims to maximize the driver's long-term cumulative reward. Besides, e-taxi drivers in fact work in a non-cooperative paradigm, where each individual e-taxi driver attempts to maximize his own reward. Thus, we model the e-taxis' decision making process as a non-cooperative Markov game [37], referred to as the *CARE game*, defined in Definition 1.

Definition 1 (CARE Game). *The CARE game is a Markov game defined by the following components.*

- Agent: *The CARE game has the set of e-taxis $\mathcal{I}$ as agents.*
- State: *A state s of the CARE game consists of the current time step t, as well as the numbers of orders, agents, and available charging posts in each grid. All states of the CARE game constitute the state space $\mathcal{S}$.*
- Observation: *At every time step t, each agent $i \in \mathcal{I}$ receives a local observation o_i consisting of its current grid, battery level, and an indicator that shows if it is serving an order. All possible observations constitute the observation space $\mathcal{O}$.*
- Action: *At every time step t, each available agent i (i.e., an e-taxi not serving any order) takes an action a_i that indicates whether it stays in the current grid or moves to one of the neighboring ones, and whether it serves an order, charges its battery, or simply stays in its next grid. We denote agent i's action space as $\mathcal{A}_i$, and the joint action space as $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$.*
- Policy: *Given observation o_i, agent i's policy π_i specifies a probability $\pi_i(a_i|o_i)$, with which it takes each action $a_i \in \mathcal{A}_i$.*
- State Transition: *Given state s and agents' joint action $\mathbf{a} = (a_1, \ldots, a_N)$, the current state s transitions to state s' according to the probability*[5] *$P(s'|s, \mathbf{a})$.*
- Reward: *At time step t, given state s and agents' joint action $\mathbf{a}$, each agent i's immediate reward is denoted as $r_{i,t}(s, \mathbf{a}) \in \{0, 1\}$ which represents the number of orders served by agent i in the current time step, and it seeks to maximize its long-term cumulative reward, denoted as $R_i = \mathbb{E}\big[\sum_{t=0}^{\infty} \alpha^t r_{i,t}\big]$, where $\alpha \in (0, 1)$ is the discount factor.*

5. Note that such randomness in state transition is caused by factors including stochastic order arrivals, etc.

Note that the CARE game is non-cooperative in nature where each agent maximizes his own reward, and thus the credit-assignment problem that oftentimes arises in multi-agent reinforcement learning [38] is naturally avoided. As the state transition and reward functions of the CARE game is *a priori* unknown in real practice, we propose to train each agent's reward-maximizing policy via our novel *multi-agent reinforcement learning* framework elaborated in Section 5.

5 Proposed MFHRL Framework

In this paper, we propose to address the CARE game and generate the reward-maximizing recommendations for e-taxi drivers via our multi-agent *mean field hierarchical reinforcement learning (MFHRL)* framework. The rest of this section first gives an overview of the proposed MFHRL framework and then describes in detail its manager and worker modules.

5.1 Framework Overview

In a real-world e-taxi environment, it is oftentimes necessary for an agent to take far-sighted actions such as charging and relocating to a distant location with potential order-serving opportunities, which are beneficial to the agent in the long run. However, in practice, a considerable amount of time usually has to be spent in the process of charging and relocating to a distant location, during which the agent will receive *no immediate reward*. Without immediate extrinsic incentives, the agents who aim to learn the reward-maximizing policies will barely explore the above charging and relocating actions, and thus can not learn effective policies to take these actions, even though these actions are beneficial to e-taxi drivers in the long run. Such a phenomenon that arises in our e-taxi environment corresponds to the notorious *sparse-reward* challenge in reinforcement learning [5], [18], which hinders traditional reinforcement learning approaches [26], [27] from effectively learning to take the aforementioned far-sighted actions.

To address this challenge, as shown in Fig. 9, each agent adopts a hierarchical MFHRL framework consisting of a *manager* module and a *worker* module similar to the existing FeUdal Networks (FuN) [5]. In MFHRL, the manager operates at a *lower temporal resolution* and can always look ahead for a fixed number of time steps. Hence, the manager is

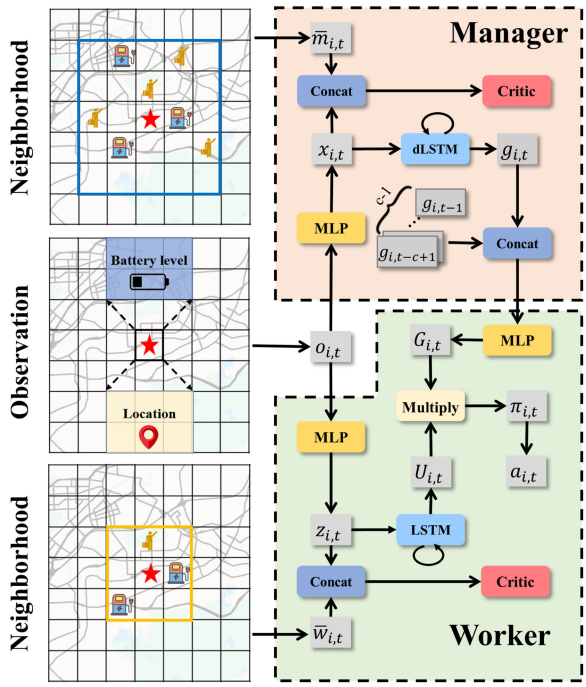


Fig. 9. The MFHRL framework of an agent (with the star indicating the location of the agent).

empowered with the ability to set goals to the worker and reward the worker for following the goals. Thus, the goals set by the manager could intrinsically guide the worker's decisions towards favorable directions. In contrast, the worker operates at a *higher temporal resolution* and outputs the agent's action, under a policy trained via both the intrinsic rewards set by the manager and extrinsic rewards obtained from the environment. As a result, such a hierarchical framework could better facilitate an agent to take far-sighted actions when necessary.

Besides, as agents may compete for the same charging or order-serving opportunities, their actions will naturally be affected by the others. In order to capture such mutual influences in a city-scale e-taxi environment with usually thousands of agents, we integrate each of the two hierarchical levels separately with the mean field approximation [6]. For each agent, such a method approximates other agents' influences on it with only the average effect of the agents in its neighborhood. Specifically, we design a mean action to represent the aforementioned average effect. As the worker focuses on a short time interval, its mean field approximation only considers the mean action of the agents within its near neighborhood. In contrast, the mean field approximation in the manager module considers the mean action of the agents in a broader area reachable by the agents within the time interval covered by the manager, which helps capture the influences of other agents' far-sighted actions. The design details for the mean action are given in the following Sections 5.2.2 and 5.3.2.

To obtain the mean action, we adopt the framework of centralized training with decentralized execution [39], allowing an agent's policy to use the mean action as one of its inputs to ease the training. However, such mean action information is not used during the execution process. To achieve this end, we employ an *advantage actor critic (A2C)* framework [39], [40] and only integrate the mean action into the critic's input.

The design details for both the manager and worker modules are given in the following Sections 5.2 and 5.3.

5.2 Manager Module

5.2.1 Architecture

As shown in Fig. 9, at each time step t, the inputs of each agent i's manager module include its observation $o_{i,t}$, and the mean action of the agents in the manager's neighborhood, denoted as $\overline{m}_{i,t}$. The observation $o_{i,t}$ is then fed into an MLP, which outputs an intermediate state representation $x_{i,t}$.

In order to enable the manager to operate at a lower temporal resolution, we adopt the *dilated LSTM (dLSTM)*, which maintains an array $(h_{i,0}, h_{i,1}, \ldots, h_{i,c-1})$ to store c historical hidden states given a dilation radius $c \in \mathbb{Z}^+$. The dLSTM takes $x_{i,t}$ and $h_{i,t\%c}$ as its input. Apart from an updated value of $h_{i,t\%c}$, the dLSTM outputs the goal $g_{i,t}$ that corresponds to time step t, which can be considered as an instruction to the worker's actions in the following c time steps. Then, the manager concatenates $c-1$ previous goals $g_{i,t-c+1}, \ldots, g_{i,t-1}$ with $g_{i,t}$. Finally, the concatenated vector is passed to the worker as its input. The reasons for such concatenation are two-fold. On one hand, these $c-1$ previous goals still contain the information that can be used to instruct the worker's actions at time step t. On the other hand, since the goal should be consistent and stable for at least a few time steps, such concatenation can make the goal delivered from the manager to the worker more consistent with the previous one.

In our MFHRL framework, the aforementioned dLSTM acts as the actor of agent i' manager, which outputs $g_{i,t}$ in each time step t as its action. Furthermore, the mean action $\overline{m}_{i,t}$, and the intermediate state representation $x_{i,t}$ are concatenated and used as the input of the manager's critic.

5.2.2 Mean Field Approximation

At time step t, the mean action $\overline{m}_{i,t}$ that agent i's manager takes as input depends on the range of the neighborhood considered by it. Clearly, the range should be defined as the areas that influence how the manager sets the goals. Given that the entire city is divided into equal-size square grids, the manager's neighborhood is thus a square area covering all grids the agent can reach within c time steps[6]. That is, we define the manager's neighborhood as a square area with $(2c+1) \times (2c+1)$ grids centered at the agent's current grid.

In this paper, for each agent i, we propose to use the *order-agent gap* and *post-agent gap* to capture the effect of other agents on it. Specifically, in each grid of the agent's neighborhood, the former refers to the gap between the number of orders and that of the agents willing to serve orders, and the latter refers to the gap between the number of available charging posts and that of the agents willing to charge. That is, $\overline{m}_{i,t}$ is a $2 \times (2c+1) \times (2c+1)$ dimensional vector that contains the order- and post-agent gaps in all grids of agent i's neighborhood. However, in real practice,

6. We assume in this paper that within one time step an agent could either stay in its current grid or move to one of the eight neighboring grids.

an agent cannot directly obtain the order- and post-agent gaps in order to calculate such mean action. As previous mentioned in Section 5.2.1, we utilize the mean action only in the training process but not in the execution process. Thus, the mean action $\overline{m}_{i,t}$ will only be used as an input of the manager's critic. The detailed training process is given in the Section 6.

5.3 Worker Module

5.3.1 Architecture

The architecture of the worker is presented in Fig. 9. At each time step t, each agent i's worker takes the local observation $o_{i,t}$ and the mean action $\overline{w}_{i,t}$ of the agents in its neighborhood as the inputs. As the manager module, an MLP is designed to transform the observation $o_{i,t}$ into an intermediate state representation $z_{i,t}$.

The mean action $\overline{w}_{i,t}$ concatenated with $z_{i,t}$ is used as the input of the critic. Different from the manager module, the worker operates at a higher temporal resolution, and only considers its action at the current time step. We thus adopt a standard LSTM in the worker module. Specifically, the hidden state $h_{i,t-1}$ of the previous time step $t-1$, and the intermediate state representation $z_{i,t}$ are fed to the LSTM, which outputs $U_{i,t}$ and the updated hidden state $h_{i,t}$.

Similar to in the manager module, the LSTM acts as the worker's actor in our MFHRL framework, which outputs $U_{i,t}$ as the preliminary action of the worker. The c concatenated goals delivered from the manager is transferred to $G_{i,t}$ through an MLP. In order to incorporate the influences of the goals set by manager, $U_{i,t}$ is further multiplied with $G_{i,t}$ and fed through a softmax layer to obtain agent i's' final policy $\pi_{i,t}$ (i.e., a probability distribution over agent i's action space $\mathcal{A}_i$). When interacting with the environment, agent i takes action $a_{i,t}$ according to the policy $\pi_{i,t}$ and receives a reward, denoted as $r_{i,t}$, from the environment.

5.3.2 Mean Field Approximation

Similar to Section 5.2.2, we define the range of the neighborhood considered by the worker as the area that influences its actions. More concretely, such a neighborhood is a square area covering all the grids that the agent can reach in one time step, i.e., 3×3 grids centered at the agent's current grid. Besides, the order-agent gap and post-agent gap are also used in the worker module to depict the influences of other agents. Thus, $\overline{w}_{i,t}$ is a $2 \times 3 \times 3 = 18$ dimensional vector consisting of the above gaps in the worker's neighborhood. As the manager module, centralized training with decentralized execution is also adopted for the worker module, and only the worker's critic takes the mean action $\overline{w}_{i,t}$ as one of its inputs.

6 Proposed Training Method

In this section, we describe in detail our method to train the manager and worker modules.

6.1 Training the Manager Module

In the training process, at each time step t, the actor of agent i's manager (i.e., dLSTM) takes $x_{i,t}$ and $h_{i,t\%c}$ as the inputs, and outputs $g_{i,t}$ as its action. The mean action $\overline{m}_{i,t}$ concatenated with $x_{i,t}$ is fed into the manager's critic, which outputs the mean field value, denoted as $V_{i,t} = V_{\phi_i}(x_{i,t}, \overline{m}_{i,t})$ where ϕ_i represents the value function's parameters. After agent i receives the reward $r_{i,t}$, it stores the tuple $\mathbf{d}_{i,t} = (x_{i,t}, \overline{m}_{i,t}, g_{i,t}, r_{i,t})$ in a replay buffer $\mathcal{D}$. We train the critic of agent i's manager by minimizing the loss given in the following:

$$\mathcal{L}(\phi_i) = \mathbb{E}_{\mathbf{d}_{i,t}\sim\mathcal{D}}\left[\left(\sum_{l=0}^{c-1}\alpha^l r_{i,t+l} + \alpha^c V_{i,t+c} - V_{i,t}\right)^2\right]. \tag{1}$$

Instead of traditional policy gradient, we adopt the *transition policy gradient* [5] method to train the manager's actor. Such method calculates the transition policy gradient on the actor as the following:

$$\nabla_{\theta_i} g_{i,t} = A^m_{i,t} \nabla_{\theta_i} d_{\cos}\left(x_{i,t+c} - x_{i,t}, g_{i,t}(\theta_i)\right), \tag{2}$$

where θ_i denotes the parameters of the actor network of agent i's manager, $d_{\cos}(a, b)$ denotes the cosine similarity between two vectors a and b and $A^m_{i,t} = R^E_{i,t} - V_{i,t}$ represents the manager's advantage function. $R^E_{i,t} = \sum_{l=0}^{\infty}\alpha^l r_{i,t+l}$ is the cumulative discounted *extrinsic rewards* that the agent obtains from the environment.

6.2 Training the Worker Module

Clearly, the goals set by the manager could lead the worker to be foresightful and take actions that benefit the agent's long-term rewards. Thus, in order to encourage the worker to take actions following the guidance of the goals, we introduce the *intrinsic reward* $r^I_{i,t}$ given in the following:

$$r^I_{i,t} = \frac{1}{c-1}\sum_{l=1}^{c-1} d_{\cos}\left(x_{i,t} - x_{i,t-l}, g_{i,t-l}\right). \tag{3}$$

In the training process, agent i aims to maximize a weighted sum of the extrinsic and intrinsic rewards $R_{i,t} = R^E_{i,t} + \lambda R^I_{i,t}$, where $\lambda \in [0, 1]$ is a hyper-parameter that controls the degree of incentivizing the worker to follow the guidance of the goals. $R^I_{i,t} = \sum_{l=0}^{\infty}\alpha^l r^I_{i,t+l}$ is the cumulative discounted intrinsic rewards that the manager rewards the agent for following the goals.

Similar to the manager module, agent i's worker takes $z_{i,t}$ and $h_{i,t-1}$ as inputs and outputs the policy $\pi_{i,t}$. The mean action $\overline{w}_{i,t}$ concatenated with $z_{i,t}$ is taken by the worker's critic as input, which then outputs the mean field value, denoted as $J_{i,t} = J_{\varphi_i}(z_{i,t}, \overline{w}_{i,t})$, where φ_i represents the value function's parameters. After taking action $a_{i,t}$ according to the policy $\pi_{i,t}$, agent i receives the reward $r_{i,t}$ at time step $t+1$. We then use a replay buffer $\mathcal{B}$ to store the tuple $\mathbf{b}_{i,t} = (z_{i,t}, \overline{w}_{i,t}, a_{i,t}, r_{i,t}, r^I_{i,t})$.

The worker's critic aims to minimize the loss function given in the following:

$$\mathcal{L}(\varphi_i) = \mathbb{E}_{\mathbf{b}_{i,t}\sim\mathcal{B}}\left[\left(r_{i,t} + \lambda r^I_{i,t} + \alpha J_{i,t+1} - J_{i,t}\right)^2\right]. \tag{4}$$

Furthermore, the actor of the worker is updated by using the policy gradient given in the following:

$$\nabla_{\vartheta_i}\pi_{i,t} = A^w_{i,t}\nabla_{\vartheta_i}\log\pi_{i,t}(a_{i,t}|z_{i,t}), \tag{5}$$

where ϑ_i denotes the parameters of the worker's actor network, and $A_{i,t} = R^E_{i,t} + \lambda R^I_{i,t} - J_{i,t}$ is the worker's advantage function.

6.3 Overall Training Algorithm

In this section, we present the overall training algorithm of MFHRL in the following Algorithm 1.

Algorithm 1. Training Algorithm of MFHRL

1 Initialize each agent i's parameters ϕ_i, θ_i, φ_i, and ϑ_i;
2 **while** *training not finished* **do**
3 **foreach** *time step t from 1 to T* **do**
4 **foreach** *agent i* **do**
5 The manager computes $x_{i,t}$ based on $o_{i,t}$;
6 The manager generates $g_{i,t}$ based on $x_{i,t}$;
7 The worker computes $z_{i,t}$ based on $o_{i,t}$;
8 The worker generates $a_{i,t}$ according to $\pi_{i,t}$;
9 The agent takes the action $a_{i,t}$ and observes $r_{i,t}$;
10 Each agent i computes $\overline{m}_{i,t}$, $\overline{w}_{i,t}$, and $r^I_{i,t}$;
11 Store each agent i's $(x_{i,t}, \overline{m}_{i,t}, g_{i,t}, r_{i,t})$ into the replay buffer $\mathcal{D}$;
12 Store each agent i's $(z_{i,t}, \overline{w}_{i,t}, a_{i,t}, r_{i,t}, r^I_{i,t})$ into the replay buffer $\mathcal{B}$;
13 **foreach** *agent i* **do**
14 Compute $V_{i,t}$ and $J_{i,t}$ for time step t from 1 to T;
15 Sample K experiences from $\mathcal{D}$ and $\mathcal{B}$ jointly;
16 Update the manager's critic using Equation (1);
17 Update the worker's critic using Equation (4);
18 Update the manager's actor using Equation (2);
19 Update the worker's actor using Equation (5);

As long as the training of MFHRL has not converged, the algorithm first collects the experiences from each agent and store them in replay buffers (line 3-12). At time step t that is smaller than a predefined threshold T, each agent i gets observation $o_{i,t}$ from the environment. The agent's manager computes the intermediate state representation $x_{i,t}$ and yields the goal $g_{i,t}$ (line 5-6). The worker then computes the intermediate state representation $z_{i,t}$ (line 7), and outputs the action $a_{i,t}$ according to the generated policy $\pi_{i,t}$ (line 8). Next, the agent takes the action $a_{i,t}$ and observes the reward $r_{i,t}$ from the environment (line 9). After that, each agent calculates the mean action $\overline{m}_{i,t}$ and $\overline{w}_{i,t}$ for the manager and worker respectively (line 10). Agents' experiences are then stored in replay buffers $\mathcal{D}$ and $\mathcal{B}$ for later training processes (line 11-12). After that, the algorithm enters the parameter updating process (line 13-19). For each agent i, the algorithm calculates its mean field values $V_{i,t}$ and $J_{i,t}$ for all the time step t from 1 to T (line 14). Then, the algorithm samples K experiences from the replay buffer $\mathcal{D}$ and $\mathcal{B}$ jointly (line 15). Finally, the parameters of the manager's and worker's actor and critic networks are updated (line 16-19).

Note that the worker and manager modules are trained separately in Algorithm 1. Although the goals delivered from the manager influence the action taken by the worker and the agent's extrinsic reward, there is no gradient transferred from the worker to the manager.

Moreover, it is also worth noting that the critic only uses the mean action information related to the policies of other agents in the training process. As long as training is completed, only the actor of each agent is employed in the execution process, which does not need the mean action as its input and operates with only the agent's local observation.

7 Experiment

In this section, we first introduce the e-taxi simulator as well as the baseline methods and the parameter settings in this paper. Then, we evaluate the performance of the proposed MFHRL framework on both the e-taxi driver and online ride-hailing platform side.

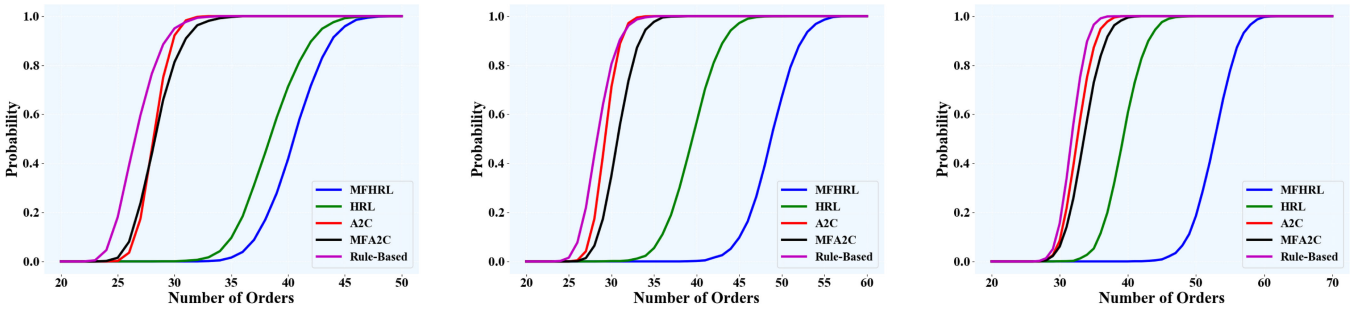
7.1 Simulator Design

To support the training and evaluation of the proposed method, we adopt and extend the grid-based simulator designed by Lin *et al.* [26] to the simulator for the dynamic urban environment of CARE.

In the grid-based simulator, we use a grid-world to represent the Shenzhen city. Specifically, the coordinate of the south-west corner of the grid-world is (latitude=22.44791, longitude=113.72469). And the coordinate of the north-east corner is (latitude=22.88723, longitude=114.53818). Given the boundary of the grid-world, we divide the Shenzhen city into 23×42 grids. After cleaning the invalid grids, e.g., the grid located in the lake, the city is covered by 702 square grids. Given GPS locations of the orders, e-taxis, and charging posts, we map these into the grids and use the grid ID to represent the location information. The driving distance between the neighbor grids is approximately 2 kilometers. And the time step is defined as 10 minutes [3].

We assume that the initial battery level of all e-taxis is as full at the first time step, and consider the battery consumption model as a function of the driving distance [1]. After that, at each time step, the agents' actions decide the distribution and battery levels of e-taxis in each grid. Likewise, we assume that all fast and slow charging posts are available for the first time step. At each time step, the agents' actions also decide the number of available fast and slow charging posts in each grid. More concretely, if e-taxi drivers decide to charge their e-taxis in one grid, we always first allocate available fast charging posts to e-taxis in a random manner. The rest e-taxi drivers who are willing to charging their e-taxis have to use the rest of the available slow charging posts. During one time step, the battery level of an e-taxi is increased by 25 percent via charging at the fast charging post. In contrast, charging at the slow charging post can only increase the e-taxi's battery level by 8 percent. Besides, by the definition of the action in Section 4, the e-taxi driver has to release the charging posts after charging for one time step.

For the order generation, we generate the orders by bootstrapping from the real-world order dataset introduced in Section 3.1 at each time step and remove the orders that are not served at the previous time step from the environment. For the order dispatching, since it is not the focus in this paper, if the e-taxis take actions and choose to serve orders, we allocate the orders for them in a random manner. When an e-taxi is serving an order, it is regarded as an unavailable agent for the simulator who cannot take the next action until it arrives at the order destination.



(a) 50% of the real-world orders. (b) 100% of the real-world orders. (c) 150% of the real-world orders.

Fig. 10. CDF of the number of orders served by an e-taxi driver in the real-world charging station scenario.

7.2 Baselines and Parameter Setting

7.2.1 Baselines

Since there is no existing work studying the charging and relocation recommendation system for a fleet of e-taxis, we take the widely used reinforcement learning frameworks as baselines. The details of these baselines are described as follows:

- *Rule-Based Method:* The heuristic policy in the rule-based method is that all the e-taxis move around randomly and aim to serve for orders as many as possible. Besides, we set the threshold of the battery level as 20 percent. If the battery level of an e-taxi is lower than this predefined threshold, it will be driven to the nearest charging station and charge its battery level to the full.
- *Independent A2C:* The independent A2C framework has been used in [26] for the multi-agent environment. More concretely, the agent in this independent A2C framework only takes its own local observation and action as the input of its critic. In our experiments, we use the RMSProp optimizer with a learning rate of 0.0001 for both the critic and the actor. The discounted factor α is set as 0.99.
- *Mean Field A2C:* Yang *et al.* [6] proposed the state-of-the-art mean field actor-critic framework. In this paper, we extend it to the mean field A2C framework. The definition of the mean action is the same as that in Section 5.3. We use a vector in the dimension of $R^{1\times 18}$ to record the mean action and integrate the mean action with the agent's observation and action into the critic's input. The hyper-parameters used in the mean field A2C framework are the same as those used in the independent A2C framework.
- *HRL:* We also take the *Hierarchical Reinforcement Learning (HRL)* in [5] as a baseline method. As described in Section 5.1, each agent involves two hierarchical modules: the manager and the worker. We set the hyper-parameter $\lambda = 0.1$ and the dilation radius $c = 3$. The RMSProp optimizer with a learning rate of 0.0003 is adopted for both the manager and the worker modules. The discounted factors for the manager and the worker are set as 0.999 and 0.99, respectively. Since HRL is designed for the single-agent environment, the agent in the HRL framework only takes its own local observation and action as the input.

7.2.2 Parameter Setting of MFHRL Framework

We set the dilation radius in our proposed MFHRL framework as $c = 3$. So the vector that records the mean action for agent i's manager is $\overline{m}_{i,t} \in R^{1\times 98}$ at time step t. The vector that records the mean action in the worker's neighborhood is $\overline{w}_{i,t} \in R^{1\times 18}$. The rest of the hyper-parameters used in the MFHRL framework are the same as those in the HRL framework.

7.3 Performance Evaluation on the E-Taxi Driver Side

In this subsection, we consider two different charging station scenarios, i.e., the real-world charging station scenario and the fast charging post scenario. In each scenario, we consider three different cases in which the order numbers are set as 50, 100, and 150 percent of the order number in the real-world order dataset, respectively. The adopted metric here is the number of the served orders in one day. The whole time horizon used in the experiments is around one month.

7.3.1 Real-World Charging Station Scenario

In the real-world charging station scenario, there are two types of charging posts: the fast charging post and the slow charging post. The fast charging post takes only 40 minutes to get an e-taxi fully charged while the slow charging post needs to take 120 minutes[7]. The experimental results in the real-world charging station scenario are shown in Fig. 10. We can observe that the proposed MFHRL framework performs the best. The reasons can be described as follows.

For the case in which the order number is 50 percent of the order number in the real-world order dataset, how to find the sparse orders is one of the major challenges for the agents. Due to the hierarchical structure, the agents with the MFHRL and HRL framework can always take far-sighted relocation actions to drive across a few grids and precisely find the potential orders. Hence, MFHRL and HRL outperform the rest of the baselines, as shown in Fig. 10a. Besides, in our design, the mean field approximation in the manager and worker module can capture the possible influences of other agents' charging and relocation actions. In contrast, HRL, which is mainly designed for a single-agent setting, fails to capture such influences. Therefore, the proposed MFHRL framework performs the best in this case.

7. https://www.byd.com/

(a) 50% of the real-world orders. (b) 100% of the real-world orders. (c) 150% of the real-world orders.

Fig. 11. CDF of the number of orders served by an e-taxi driver in the fast charging post scenario.

With the number of orders increases, the agents probably can serve more orders but consume more battery at the same time. Given the limited number of charging posts, competitions for the charging opportunities will be intensified. Additionally, the agents may still compete for the same order-serving opportunities. Hence, how to incorporate the agents' mutual influences in the decision making process becomes crucial for all agents. In our design, we integrate each hierarchical level of MFHRL separately with the mean field approximation to incorporate e-taxis' mutual influences in decision making. The agents with the MFHRL framework can capture the possible influences caused by other agents' far-sighted actions. At the same time, they can also effectively avoid unnecessary local competitions. Thus, as shown in Figs. 10b and 10c, the proposed MFHRL framework still performs the best.

Moreover, we can observe in Figs. 10 that A2C and MFA2C cannot achieve good performance in all the three cases in the real-world charging station scenario. This is mainly because the two methods cannot effectively guide the agents to take far-sighted actions due to the lack of a hierarchical structure. More specifically, we find out that A2C and MFA2C scarcely instruct the agents to charge their e-taxis in advance to avoid charging during the time steps with more order-serving opportunities. As a result, A2C and MFA2C fail to learn effective charging policies.

7.3.2 Fast Charging Post Scenario

With the rapid development of charging technology, the slow charging posts will be finally replaced by the fast charging posts in the future [10]. In this experiment, we consider a scenario in which there are only fast charging posts, and we achieve this by replacing all the slow charging posts with the fast charging ones. The experimental results in the fast charging post scenario are shown in Fig. 11. Even though the charging time is greatly reduced in this scenario, we can still observe in Fig. 11 that the proposed MFHRL framework performs much better than all the baselines.

In Fig. 11, it is worth noting that the performance of the rule-based method is close to that of MFA2C in all three cases. Worse still, the performance of A2C is even worse than that of the rule-based method. This result demonstrates that due to the lack of a hierarchical structure, the policies learned by A2C and MFA2C in an urban e-taxi environment are even less effective than the heuristic policies set by the rule-based method.

In contrast, as shown in Fig. 11, HRL and MFHRL can still perform much better than the rest of the baselines. This is mainly because the hierarchical structure of HRL and MFHRL can enable agents to learn effective charging and relocation policies. Furthermore, we can observe that the performance of MFHRL is also much better than that of HRL. The reason is that the mean-field approximation in the manager and worker module can effectively incorporate e-taxis' mutual influences in decision making.

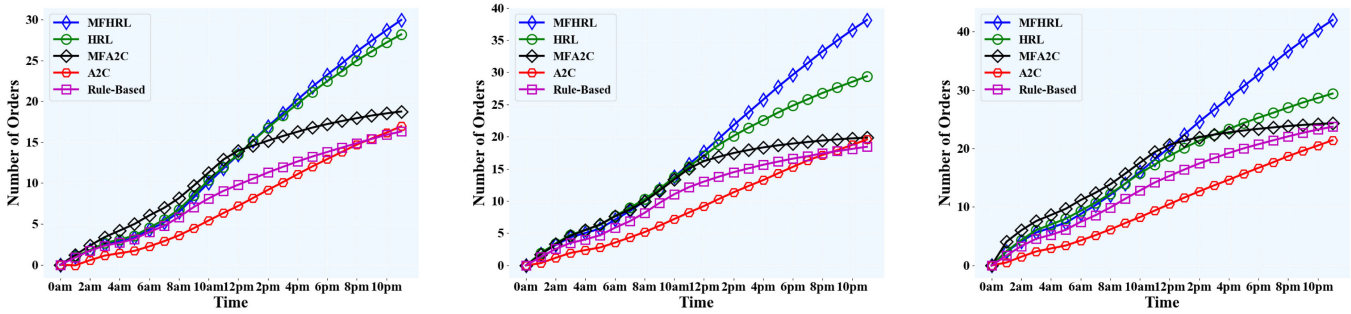
7.3.3 Case Study

We conduct case studies to further evaluate the performance of the proposed MFHRL and demonstrate the effectiveness of the charging and relocation policies derived by MFHRL. Considering that all agents have the same initial state at the first time step, we use this state as the initial state in our case study and evaluate the one-day performance.

As shown in Fig. 16a, we can observe that MFHRL and HRL outperform the rest of the baselines. The reason is that the hierarchical structure of MFHRL and HRL enables the agents to oftentimes take far-sighted relocation actions to precisely find the sparse orders. Even for the case in which the order number is just 50 percent of the order number in the real-world order data set, such a hierarchical structure can still ensure a steady increase in the average cumulative rewards as time passes.

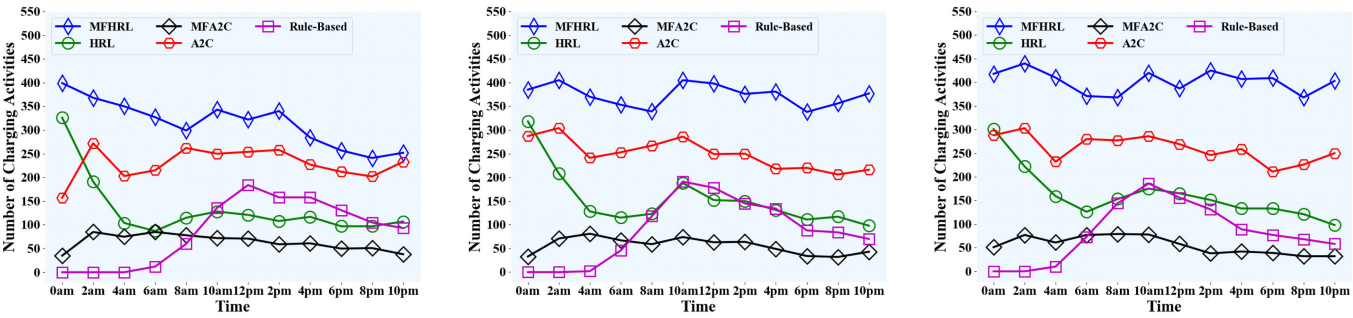
As the total number of orders in the scenario increases, the competitions on the same order-serving opportunities are intensified. As shown in Fig. 12, it is worth noting that the performance gap between our proposed MFHRL and the baselines is also amplified. More concretely, we can observe that, for the performance of MFHRL, the average number of served orders by one e-taxi driver is 29, 38, and 43 for the cases of 50, 100, and 150 percent of the real-world orders. In contrast, the performance of HRL almost keep unchanged in three cases. The average number of served orders in three cases is around 28. The reason is that, compared with the policies of HRL, the policies of our proposed MFHRL takes account of mutual influences among agents. Hence, the agents who follow the recommendations provided by the MFHRL can effectively deal with the intensive competitions and avoid unnecessary competitions on the same order-serving opportunities.

Besides, as shown in Fig. 12, we can observe that, for the performance of MFA2C, the number of served orders increases from 17 shown in Fig. 16a to 23 shown in Fig. 12c.



(a) 50% of the real-world orders. (b) 100% of the real-world orders. (c) 150% of the real-world orders.

Fig. 12. E-taxi driver's average cumulative rewards within one day in the real-world charging station scenario.



(a) 50% of the real-world orders. (b) 100% of the real-world orders. (c) 150% of the real-world orders.

Fig. 13. The number of charging activities in the real-world charging station scenario.

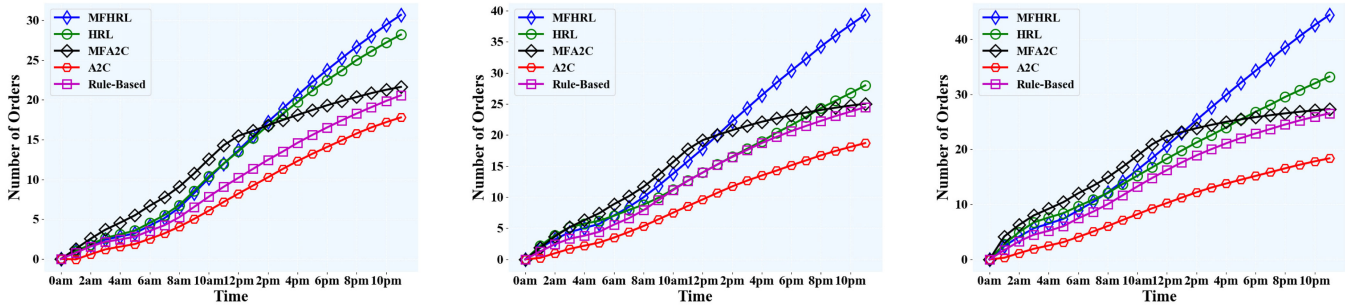
However, such a performance is much worse than that of MFHRL. The reason is two-fold. On one hand, due to the hierarchical structure, MFHRL can enable agents to learn far-sighted charging and relocation actions to explore the available charging posts and orders. On the other hand, instead of merely concentrating on the local competition as MFA2C does, apart from the local mutual influences, the policies of our proposed MFHRL also consider the mutual influences caused by the other agents' far-sighted actions.

Moreover, we can observe in Fig. 12 that our proposed MFHRL can ensure a steady increase in the average cumulative rewards for all three cases in a real-world charging station scenario. One of the primary reasons is on the charging policy learned by the MFHRL. To analysis the learned charging policy, we redefine the charging activity in this section. For details, if an agent takes a charging action and successfully charges his e-taxi at the charging station, we regard such a process as a charging activity. Based on the e-taxi simulator described in Section 7.1, each charging activity in this section will last for 10 minutes.

From Fig. 13, we can observe that the number of charging activities taken by the agents of MFHRL is the largest. The first reason is that the charging policy learned by the MFHRL can avoid unnecessary competitions and grasp every available charging opportunity. Hence, when taking a charging action, the agent of MFHRL has the highest probability to successfully charge his e-taxi. In addition, charging policy learned by MFHRL can oftentimes guide the agent to take a far-sighted charging action. More concretely, instead of only charging the e-taxis when its battery is running low, the agents of MFHRL can oftentimes charge their e-taxi in advance. For details, when charging activities happen, the average remaining battery levels of agents of MFHRL are 49.23, 44.17, and 37.17 percent for the cases of 50, 100, and 150 percent of the order numbers, respectively. Furthermore, we also find out that the charging policy learned by MFHRL can often choose the reasonable charging time step to avoid charging during the time steps with more order-serving opportunities.

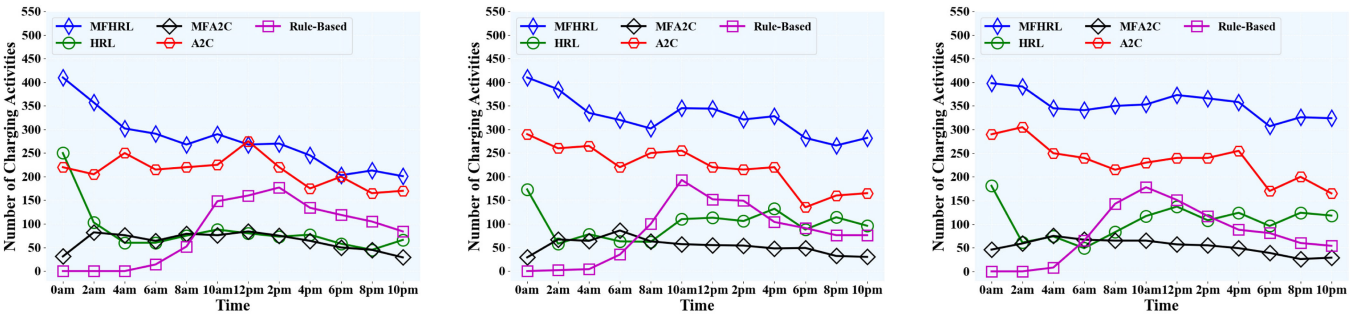
In contrast, the charging policies of the baselines are not as effective as that of our proposed MFHRL is. For the HRL framework, since the learned charging policy does not incorporate agents' mutual influences in decision making, the probability that the agent of HRL can successfully charge his e-taxi when taking a charging action is just 65.31 percent. As shown in Fig. 13, the number of charging activities taken by the agents of HRL is much smaller than that taken by the agents of MFHRL. Besides, in our simulator, when an e-taxi depletes its battery in a grid, the e-taxi driver has to stay at that location forever and thus cannot obtain any reward afterwards. To avoid depleting the battery, the agent of A2C learns a conservative charging policy, which frequently guides the agent of A2C to charge his e-taxi, as shown in Fig 13. Such a conservative charging policy greatly reduces the order-serving duration. Therefore, we can observe that the performance of A2C is even worse than that of the rule-based method, as shown in Fig. 12c.

Figs. 14 and 15 show the experimental results on average cumulative rewards and the number of charging activities in a fast charging scenario. It is worth noting that due to the effective charging and relocation policy, our proposed MFHRL also keeps a steady increase in the average cumulative rewards in a fast charging post scenario. Additionally, in Figs. 14 and 15, we can also observe that the average



(a) 50% of the real-world orders. (b) 100% of the real-world orders. (c) 150% of the real-world orders.

Fig. 14. E-taxi driver's average cumulative rewards within one day in the fast charging post scenario.



(a) 50% of the real-world orders. (b) 100% of the real-world orders. (c) 150% of the real-world orders.

Fig. 15. The number of charging activities in the fast charging post scenario.

cumulative rewards of MFHRL at the end of the day have greatly surpassed that of other baselines.

7.4 Performance Evaluation on the Online Ride-Hailing Platform Side

Nowadays, most taxi drivers are the users of online ride-hailing platforms, such as Uber and Didi Chuxing. Thus, our proposed recommendation system CARE can be implemented on these platforms and used to boost the experience of e-taxi drivers. In this subsection, we show that, apart from the improvement of e-taxi drivers' experience, the proposed MFHRL framework can also ensure satisfactory global rewards for the platforms. In Section 7.3, we evaluate the performance in two different charging station scenarios with three different order-number cases. More specifically, the adopted metrics here are *Order Response Rate (ORR)* and *Gross Merchandise Volume (GMV)*. These two metrics are the major objectives of online ride-hailing platforms. ORR is defined as the ratio between the number of served orders and that of all the orders received by the platforms. GMV represents the total incomes obtained from the served orders.

Tables 2 and 3 show the relative differences between the GMV (and ORR) of the corresponding reinforcement learning framework and that of the rule-based method for the two scenarios. We can observe that our proposed MFHRL on both GMV and ORR performs much better than that of other reinforcement learning frameworks. The reasons are as follows. First of all, the proposed MFHRL incorporates other agents' mutual influences in decision making. Thus, such a framework can avoid unnecessary competitions on the same charging and order-serving opportunities among a fleet of e-taxis. Second, instead of using virtual rewards to motivate agents to balance the demand-supply gap for the online ride-hailing platforms [27], the agents of MFHRL can spontaneously find the potential demand-supply gaps and drive across a few grids to serve orders so as to maximize

TABLE 2
Performance Comparison in the Real-World Charging Station Scenario

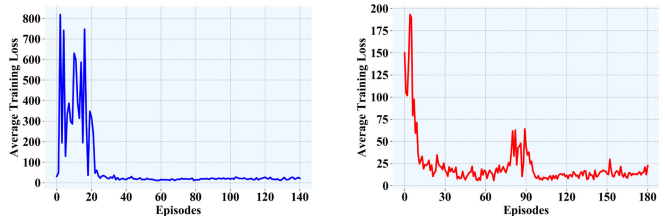
	50%		100%		150%	
	GMV	ORR	GMV	ORR	GMV	ORR
A2C	10.39%	7.26%	7.73%	1.75%	0.14%	−6.57%
MFA2C	27.55%	19.02%	17.63%	7.60%	9.94%	−0.36%
HRL	59.75%	44.65%	55.91%	40.35%	31.92%	21.53%
MFHRL	85.16%	69.65%	94.81%	73.39%	88.54%	67.88%

This table reports the relative differences between the GMV (and ORR) of the corresponding reinforcement learning framework and that of the rule-based method.

TABLE 3
Performance Comparison in the Fast Charging Post Scenario

	50%		100%		150%	
	GMV	ORR	GMV	ORR	GMV	ORR
A2C	−16.08%	−18.65%	−14.91%	−14.36%	−16.93%	−12.00%
MFA2C	17.34%	17.06%	20.68%	13.52%	15.46%	9.45%
HRL	23.07%	16.67%	27.23%	21.97%	15.27%	20.36%
MFHRL	27.97%	32.54%	45.50%	32.68%	29.19%	37.09%

This table reports the relative differences between the GMV (and ORR) of the corresponding reinforcement learning framework and that of the rule-based method.



(a) The real-world charging post scenario.

(b) The fast charging post scenario.

Fig. 16. The order number in the scenario is set as 100 percent of the order number in the real-world order dataset.

their long-term reward. Last but not least, the charging policy learned by MFHRL can always choose the reasonable charging time step to avoid charging during the time steps with more order-serving opportunities. Therefore, although our proposed MFHRL takes the perspective of e-taxi drivers, it can still ensure a satisfactory global ORR and GMV.

7.5 Convergence of the Training Algorithm

During the training process, we consider that the training algorithm of the proposed MFHRL framework achieves convergence when the loss maintains a relatively small value for at least 20 episodes. The losses considered in this paper include the mean square error loss in Equations (1) and (4) as well as the policy gradient loss in Equation (2) and 5. Fig. 16 shows examples of the convergence of the proposed algorithm in two different charging post scenarios whose order number is set as 100 percent of the order number in the real-world order dataset. Specifically, we sum up all the above losses in Equations (1), (2), (4), and (5) and calculate the average value of the summation for each agent in each step within one episode. As shown in Fig. 16, we use the average training loss to represent the aforementioned summation of losses. It is worth noting that, as shown in both subfigures, the average training loss is smaller than 30 after 100 episodes. Thus, we consider that the training algorithm has achieved the convergence in these two scenarios.

8 Discussion

In this section, we first show that, apart from ensuring satisfactory rewards for e-taxi drivers, our proposed recommendation system CARE can greatly reduce their range anxiety as well. Then, we discuss why CARE can be easily implemented on the existing online ride-hailing platforms.

8.1 Reduction on Range Anxiety

Range anxiety [1] is the fear of an electric vehicle (EV) driver that the battery of his vehicle is running low before reaching the destination or an available charging post. Range anxiety has been regarded as one of the primary barriers to the wide deployment of EVs. Worse still, compared with private EV drivers, e-taxi drivers have to drive for a longer daily distance, which consumes more battery. Thus, range anxiety is inevitably an issue faced by e-taxi drivers as well.

However, e-taxi drivers' range anxiety could be greatly reduced by simply following the charging recommendations provided by CARE. The reasons are as follows. First, the charging policy learned by MFHRL can oftentimes choose

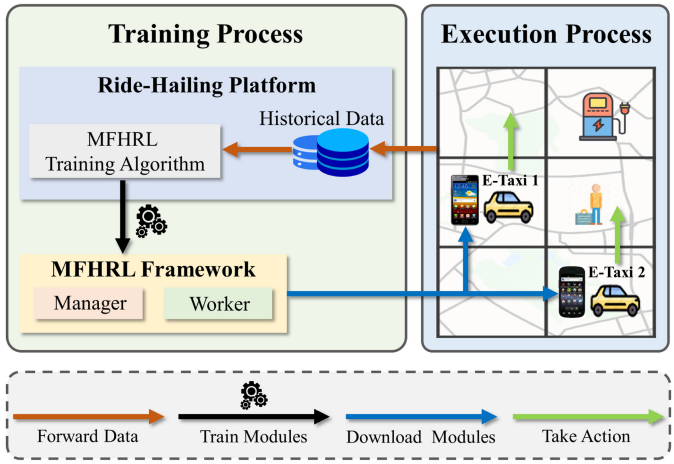


Fig. 17. The implementation of MFHRL on the existing online ride-hailing platforms.

the reasonable charging time steps and enable e-taxi drivers to charge their e-taxis in advance to avoid charging during the time steps with more order-serving opportunities. As described in Section 7.3.3, when the e-taxi drivers follow the charging recommendations of CARE, the average remaining battery levels of these e-taxis are 49.23, 44.17, and 37.17 percent for the cases of 50, 100, and 150 percent of the order numbers in practice. Second, the charging policy learned by the MFHRL has considered the mutual influences of other e-taxis. Thus, when choosing a charging action, MFHRL ensures the highest probability of choosing a charging station with available charging posts among all the methods. Last but not least, as described in Section 7.3.3, in our simulator, when an e-taxi depletes its battery, the e-taxi driver cannot obtain any reward afterwards, since he has to stay at that location forever. Clearly, the reward-maximizing policy of MFHRL has already been trained to avoid such an embarrassing situation.

8.2 Implementation on the Existing Online Ride-Hailing Platforms

As described in Section 6.4, we show that the proposed MFHRL framework can ensure a satisfactory reward for the online ride-hailing platforms. In this section, we first give an implementation overview of the proposed MFHRL framework. Then, in this subsection, we will further discuss two potential and practical implementation requirements of such platforms and demonstrate why our proposed recommendation system CARE can meet the requirements.

8.2.1 Implementation Overview

We adopt the framework of centralized training with decentralized execution in this paper. As shown in Fig. 17, on the online ride-hailing platform side, we implement the proposed MFHRL training algorithm on its servers and train a centralized MFHRL framework consisting of both manager and worker modules by using the historical data. On the e-taxi driver side, the ride-hailing platform's application only downloads the decentralized parameter-shared actor networks in the manager and worker modules, which takes the e-taxi driver's local observation as an input and outputs an

action as the recommendation. After the drivers take the corresponding actions, the application will forward data on their observations and actions to the database in the online ride-hailing platform.

8.2.2 Implementation in Other Cities

The existing online ride-hailing platforms, such as Uber and Didi Chuxing, focus on providing the worldwide service. Hence, such platforms will naturally expect that our proposed recommendation system CARE could be implemented in the different dynamic urban environments and meanwhile ensure the satisfactory performance. In this paper, though we only use the real-world e-taxi datasets in Shenzhen to set up the simulator, we evaluate the proposed MFHRL framework in 6 different dynamic urban environments, which already cover a wide spectrum of real-world scenarios. The experimental results show that the proposed MFHRL framework can greatly outperform all baselines and significantly increase the obtained rewards of e-taxi drivers. Besides, the training process of the proposed MFHRL framework for a new urban environment is not costly. In this paper, we use one GTX Titan Xp GPU with 12 TFLOPS. The training process of MFHRL under such an average hardware setting will cost at most one day.

8.2.3 Requirement on Running Time

After finishing the actions recommended by the online ride-hailing platforms, the e-taxi drivers that use such platforms expect to obtain the next recommendation shortly. Hence, the online ride-hailing platforms generally have the requirement for the running time. In this paper, we adopt the framework of centralized training with decentralized execution. The size of the proposed MFHRL framework is 110 KB consisting of 9 fully connected layers, 1 dilated LSTM, and 1 LSTM, as well as 1 convolutional neural network. During the training process, the total number of the learnable parameters in the MFHRL framework is 22275. In contrast, during the execution process, we only employ decentralized parameter-shared actor networks. Thus, given the e-taxi driver's local observation as an input, the actor networks can output an action as the recommendation at the millisecond level.

9 Conclusion

In this paper, to boost the user experience of e-taxis drivers, we design a charging and relocation recommendation system for e-taxi drivers, named as CARE. We take the perspective of e-taxi drivers and formulate their decision making as a multi-agent reinforcement learning problem, where each e-taxi driver aims to maximize his own cumulative rewards. To solve this problem, we propose a novel *multi-agent mean field hierarchical reinforcement learning (MFHRL)* framework. To the best of our knowledge, MFHRL is the first multi-agent reinforcement learning framework that integrates hierarchical reinforcement learning with mean field approximation. More specifically, the hierarchical architecture of MFHRL can set goals for the agents to effectively learn far-sighted charging and relocation decisions. Additionally, integrating each of the two hierarchical levels separately with the mean field approximation can effectively incorporate agents' mutual influences in decision making. The extensive experiments are conducted with one of the largest real-world e-taxi datasets, which contains the GPS trajectory data and transaction data of 3848 e-taxis from June 1st to June 30th, 2017, coupled with 165 charging stations including 317 fast charging posts and 1421 slow charging posts. We validate that the proposed MFHRL framework can greatly outperform all baselines and significantly increase the obtained rewards of e-taxi drivers. Besides, we provide a solid discussion on the charging policy learned by the MFHRL framework and illustrate that the learned charging policy can effectively reduce the range anxiety of e-taxi drivers. Furthermore, we also demonstrate that the MFHRL framework can ensure a satisfactory reward for online ride-hailing platforms. For the wide deployment of our proposed recommendation system CARE, we discuss CARE is readily implementable on the existing online ride-hailing platforms.

Acknowledgments

This work was supported in part by the National Key R&D Program of China 2018AAA0101200, in part by NSF China 61902244, 61960206002, 62041205, and 61829201, in part by the Shanghai Municipal Science and Technology Commission Grant 19YF1424600, in part by the US National Science Foundation under Grants CNS-1737590 and 1626374.

References

[1] A. Sarker, H. Shen, and J. A. Stankovic, "MORP: Data-driven multi-objective route planning and optimization for electric vehicles," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2018, pp. 162–197.

[2] G. Wang *et al.*, "sharedcharging: Data-driven shared charging for large-scale heterogeneous electric vehicle fleets," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2019, pp. 1–25.

[3] L. Yan *et al.*, "Employing opportunistic charging for electric taxi-cabs to reduce idle time," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2018, pp. 1–25.

[4] G. Wang, Y. Zhang, Z. Fang, S. Wang, F. Zhang, and D. Zhang, "Faircharge: A data-driven fairness-aware charging recommendation system for large-scale electric taxi fleets," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2020, pp. 1–25.

[5] A. S. Vezhnevets *et al.*, "Feudal networks for hierarchical reinforcement learning," in *Proc. 34th ACM Int. Conf. Mach. Learn.*, 2017, pp. 3540–3549.

[6] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," in *Proc. 35th ACM Int. Conf. Mach. Learn.*, 2018, pp. 5567–5576.

[7] E. Kim, J. Lee, and K. G. Shin, "Modeling and real-time scheduling of large-scale batteries for maximizing performance," in *Proc. 36th IEEE Real-Time Syst. Symp.*, 2015, pp. 33–42.

[8] G. Wang, X. Xie, F. Zhang, Y. Liu, and D. Zhang, "bCharge: Data-driven real-time charging scheduling for large-scale electric bus fleets," in *Proc. 39th IEEE Real-Time Syst. Symp.*, 2018, pp. 45–55.

[9] Y. Yuan, D. Zhang, F. Miao, J. Chen, T. He, and S. Lin, "p^{2}charging: Proactive partial charging for electric taxi systems," in *Proc. 39th IEEE Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 688–699.

[10] G. Wang, X. Chen, F. Zhang, Y. Wang, and D. Zhang, "Experience: Understanding long-term evolving patterns of shared electric vehicle networks," in *Proc. 25th ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2019, pp. 1–12.

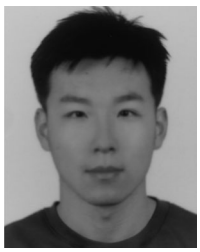
[11] F. Kong, Q. Xiang, L. Kong, and X. Liu, "On-line event-driven scheduling for electric vehicle charging via park-and-charge," in *Proc. 37th IEEE Real-Time Syst. Symp.*, 2016, pp. 69–78.

[12] F. Kong and X. Liu, "Distributed deadline and renewable aware electric vehicle demand response in the smart grid," in *Proc. 36th IEEE Real-Time Syst. Symp.*, 2015, pp. 23–32.

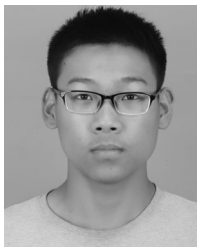
[13] G. Wang, F. Zhang, and D. Zhang, "tCharge-a fleet-oriented real-time charging scheduling system for electric taxi fleets," in *Proc. 17th ACM Conf. Embedded Netw. Sensor Syst.*, 2019, pp. 440–441.

[14] L. Yan, H. Shen, J. Zhao, C. Xu, F. Luo, and C. Qiu, "Catcharger: Deploying wireless charging lanes in a metropolitan road network through categorization and clustering of vehicle traffic," in *Proc. 36th IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.
[15] B. Du, Y. Tong, Z. Zhou, Q. Tao, and W. Zhou, "Demand-aware charger planning for electric vehicle sharing," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1330–1338.
[16] Y. Li, J. Luo, C.-Y. Chow, K.-L. Chan, Y. Ding, and F. Zhang, "Growing the charging station network for electric vehicles with trajectory data analytics," in *Proc. 31st IEEE Int. Conf. Data Eng.*, 2015, pp. 1376–1387.
[17] C. Liu, K. Deng, C. Li, J. Li, Y. Li, and J. Luo, "The optimal distribution of electric-vehicle chargers across a city," in *Proc. 16th IEEE Int. Conf. Data Mining*, 2016, pp. 261–270.
[18] O. Nachum, S. S. Gu, H. Lee, and S. Levine, "Data-efficient hierarchical reinforcement learning," in *Proc. 31st ACM Advances Neural Inf. Process. Syst.*, 2018, pp. 3303–3313.
[19] Y. Zhang, J. Chen, L. Cai, and J. Pan, "EV charging network design with transportation and power grid constraints," in *Proc. 37th IEEE Conf. Comput. Commun.*, 2018, pp. 2492–2500.
[20] Z. Dong, C. Liu, Y. Li, J. Bao, Y. Gu, and T. He, "REC: Predictable charging scheduling for electric taxi fleets," in *Proc. 38th IEEE Real-Time Syst. Symp.*, 2017, pp. 287–296.
[21] H. Yi, Q. Lin, and M. Chen, "Balancing cost and dissatisfaction in online EV charging under real-time pricing," in *Proc. 38th IEEE Conf. Comput. Commun.*, 2019, pp. 1801–1809.
[22] R. S. Sutton *et al.*, *Introduction to Reinforcement Learning*. vol. 2, Cambridge, MA, USA: MIT Press, 1998.
[23] S. Ji, Y. Zheng, Z. Wang, and T. Li, "A deep reinforcement learning-enabled dynamic redeployment system for mobile ambulances," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2019, pp. 1–20.
[24] L. Wang, X. Geng, X. Ma, D. Zhang, and Q. Yang, "Ridesharing car detection by transfer learning," *Artif. Intell.*, vol. 273, pp. 1–18, 2019.
[25] S. Guo *et al.*, "ROD-revenue: Seeking strategies analysis and revenue prediction in ride-on-demand service using multi-source urban data," *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2202–2220, Sep. 2020.
[26] K. Lin, R. Zhao, Z. Xu, and J. Zhou, "Efficient large-scale fleet management via multi-agent deep reinforcement learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1774–1783.
[27] M. Li *et al.*, "Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning," in *Proc. 28th ACM World Wide Web Conf.*, 2019, pp. 983–994.
[28] X. Tang *et al.*, "A deep value-network based approach for multi-driver order dispatching," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 1780–1790.
[29] J. Jin *et al.*, "Coride: Joint order dispatching and fleet management for multi-scale ride-hailing platforms," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 1983–1992.
[30] L. Zhang *et al.*, "A taxi order dispatch model based on combinatorial optimization," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 2151–2159.
[31] M. Xu, D. Wang, and J. Li, "DESTPRE: A data-driven approach to destination prediction for taxi rides," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2016, pp. 729–739.
[32] X. Xie, F. Zhang, and D. Zhang, "Privatehunt: Multi-source data-driven dispatching in for-hire vehicle systems," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2018, pp. 1–26.
[33] S. Guo, C. Chen, Y. Liu, K. Xu, and D. M. Chiu, "Modelling passengers' reaction to dynamic prices in ride-on-demand services: A search for the best fare," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2018, pp. 1–23.
[34] S. Guo *et al.*, "A simple but quantifiable approach to dynamic price prediction in ride-on-demand services leveraging multi-source urban data," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2018, pp. 1–24.
[35] T. Kumsila and S. Phithakkitnukoon, "Jerney: A peer-to-peer shared public transit on fixed routes," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2018, pp. 1060–1068.
[36] F. Zhang, N. J. Yuan, D. Wilkie, Y. Zheng, and X. Xie, "Sensing the pulse of urban refueling behavior: A perspective from taxi mobility," *ACM Trans. Intell. Syst. Technol.*, vol. 15, pp. 1–23, 2015.
[37] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. 11th Int. Conf. Int. Conf. Mach. Learn.*, 1994, pp. 157–163.
[38] D. T. Nguyen, A. Kumar, and H. C. Lau, "Credit assignment for collective multiagent rl with global rewards," in *Proc. 32nd ACM Advances Neural Inf. Process. Syst.*, 2018, pp. 8102–8113.
[39] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 31st ACM Advances Neural Inf. Process. Syst.*, 2017, pp. 6379–6390.
[40] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd ACM Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

Enshu Wang received the MS degree in communication and information technology from Jilin University, Changchun, China. He is currently working toward the PhD degree in computer science and engineering at SUNY Buffalo. His current research interests include multi-agent reinforcement learning, game theory and the applications on connected and autonomous vehicles.

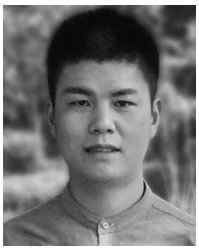


Rong Ding is currently working toward the BS degree in computer science and engineering at Shanghai Jiao Tong University, China. His current research interests include reinforcement learning and its applications in networks and systems.



Zhaoxing Yang received the BS degree in computer science and technology from the Huazhong University of science and technology, Wuhan, China, in 2019. He is currently working toward the MS degree in computer science at Shanghai Jiao Tong University, Shanghai, China. His current research interests include reinforcement learning, multi-agent system and game theory.

Haiming Jin (Member, IEEE) received the BS degree from Shanghai Jiao Tong University, Shanghai, China, in 2012, and the PhD degree from the University of Illinois at Urbana–Champaign (UIUC), Urbana, IL, in 2017. He is currently a tenure-track assistant professor at the John Hopcroft Center for Computer Science and the Department of Electronic Engineering, Shanghai Jiao Tong University. Before this, he was a postdoctoral research associate with the Coordinated Science Laboratory, UIUC. His research interests include addressing unfolding research challenges in the general areas of urban computing, cyber-physical systems, crowd and social sensing systems, network economics and game theory, reinforcement learning, and mobile pervasive and ubiquitous computing.

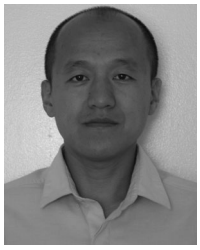


Chenglin Miao (Member, IEEE) received the PhD degree in computer science and engineering from the State University of New York at Buffalo, in 2020. He is currently an assistant professor at the Department of Computer Science, University of Georgia. His research interests include security and privacy, Internet of Things, and machine learning. He is a member of the ACM.

Lu Su (Member, IEEE) received the MS degree in statistics and PhD degree in computer science from the University of Illinois at Urbana-Champaign, in 2012 and 2013, respectively. He is currently an associate professor at the Department of Computer Science and Engineering, SUNY Buffalo. His research interests include the general areas of mobile and crowd sensing systems, Internet of Things, and cyber-physical systems. He has also worked at the IBM T. J. Watson Research Center and National Center for Supercomputing Applications. He is the recipient of the NSF CAREER Award, the University at Buffalo Young Investigator Award, the ICCPS'17 Best Paper Award, and the ICDCS'17 Best Student Paper Award. He is a member of the ACM.

Fan Zhang (Member, IEEE) received the PhD degree in communication and information system from the Huazhong University of Science and Technology, in 2007. He was a postdoctoral fellow with the University of New Mexico and with the University of Nebraska-Lincoln from 2009 to 2011. He is currently a professor at the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He is also the director of the Shenzhen Institute of Beidou Applied Technology (SIBAT). His research topics include intelligent transportation systems, urban computing, and AI technology.

Chunming Qiao (Fellow, IEEE) is currently a SUNY distinguished professor and also the current chair of the Computer Science and Engineering Department, University at Buffalo. He was elected to IEEE fellow for his contributions to optical and wireless network architectures and protocols. His current research interests include connected and autonomous vehicles. He has published extensively with an h-index of more than 69 (according to Google Scholar). Two of his papers have received the best paper awards from IEEE and Joint ACM/IEEE venues. He also has seven US patents and served as a consultant for several IT and Telecommunications companies since 2000. His research has been funded by a dozen major IT and telecommunications companies including Cisco and Google, and more than a dozen NSF grants.



Xinbing Wang (Senior Member, IEEE) received the BS degree (with hons.) in automation from Shanghai Jiao Tong University, Shanghai, China, in 1998, the MS degree in computer science and technology from Tsinghua University, Beijing, China, in 2001, and the PhD degree with a major in electrical and computer engineering and minor in mathematics from North Carolina State University, Raleigh, in 2006. Currently, he is a professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. His research interests include resource allocation and management in mobile and wireless networks, TCP asymptotics analysis, wireless capacity, cross-layer call admission control, asymptotics analysis of hybrid systems, and congestion control over wireless ad hoc and sensor networks. He has been a member of the Technical Program Committees of several conferences including ACM MobiCom 2012, ACM MobiHoc 2012, and IEEE INFOCOM 2009-2013.