Rebalancing Expanding EV Sharing Systems with Deep Reinforcement Learning

Man Luo[1,2] , **Wenzhe Zhang**[3†] , **Tianyou Song**[3†] , **Kun Li**[3†] , **Hongming Zhu**[3] , **Bowen Du**[1] and **Hongkai Wen**[1*]
[1]Department of Computer Science, University of Warwick, UK
[2]The Alan Turing Institute, UK
[3]School of Software Engineering, Tongji University, China
{m.luo.1, b.du, hongkai.wen}@warwick.ac.uk,
{zhangwenzhe, 1551177, likun, zhu_hongming}@tongji.edu.cn

Abstract

Electric Vehicle (EV) sharing systems have recently experienced unprecedented growth across the world. One of the key challenges in their operation is vehicle rebalancing, i.e., repositioning the EVs across stations to better satisfy future user demand. This is particularly challenging in the shared EV context, because i) the range of EVs is limited while charging time is substantial, which constrains the rebalancing options; and ii) as a new mobility trend, most of the current EV sharing systems are still continuously expanding their station networks, i.e., the targets for rebalancing can change over time. To tackle these challenges, in this paper we model the rebalancing task as a Multi-Agent Reinforcement Learning (MARL) problem, which directly takes the range and charging properties of the EVs into account. We propose a novel approach of policy optimization with action cascading, which isolates the non-stationarity locally, and use two connected networks to solve the formulated MARL. We evaluate the proposed approach using a simulator calibrated with 1-year operation data from a real EV sharing system. Results show that our approach significantly outperforms the state-of-the-art, offering up to 14% gain in order satisfied rate and 12% increase in net revenue.

1 Introduction

Recently, shared E-mobility systems have been expanding fast in major cities around the world [Shaheen *et al.*, 2018]. They provide a convenient way for users to pick up shared Electric Vehicles (EVs) from nearby stations and drive around whenever needed, which is a more sustainable mobility paradigm that can effectively reduce the numbers of vehicles on the roads as well as cutting out unnecessary emissions. They could also bring significant societal benefits as they offer a much more affordable and efficient on-demand mobility option to the public than traditional taxi or car renting.

Despite their rapid growth, a major problem of current EV sharing systems is the imbalanced distribution of their EV fleets as they operate over time. For instance, in morning rush hours a large volume of EVs tend to flow to central areas and stay there, making few or even no vehicles available in other places. This certainly deteriorates the overall performance, as potential customers may refrain from using the system if there are no available EVs, or no parking spaces near their destinations. It may also have long-term impact, leading to skewed vehicle distributions, e.g. some "hot" areas may accumulate substantially more EVs than others. Thus rebalancing the vehicle distributions of the system is a vital task.

In fact this is very common in shared mobility systems, e.g., shared bikes [Ghosh *et al.*, 2017; Li *et al.*, 2018; Ghosh *et al.*, 2016b], taxi [Lin *et al.*, 2018; Li *et al.*, 2019; Wei *et al.*, 2017], and ride-sharing services [Kooti *et al.*, 2017; Jiang *et al.*, 2018]. However, unlike those systems, the rebalancing problem in EV sharing has two unique challenges. First of all, EVs typically have limited range, and the charging time is much longer than filling up traditional vehicles. This adds many implicit constraints, e.g. EVs can't be repositioned to locations that are beyond their remaining range, and they also need to be sufficiently charged to serve future user orders. Secondly, as the concept of EV sharing systems is relatively new, at this stage they tend to continuously expand their infrastructure. For instance, the EV sharing system studied in this paper has doubled its stations within just 12 months, where stations are being deployed/closed every day. This makes the rebalancing task even more challenging, as at each time the candidate stations to which EVs may be repositioned are dynamically changing.

To address these challenges, in this paper we propose a novel user-incentive rebalancing approach based on multi-agent reinforcement learning (MARL), which offers monetary rewards to the end users, incentivizing them to reposition EVs to the desired locations. To tackle the challenges of limited EV range and the charging delays, we propose to incorporate the range and charging information directly in our MARL algorithm, so that the agents are fully aware of those constrains when making decisions. To cope with continuous system expansion, we propose a new action cascading approach, which decomposes the action of repositioning an EV into two subsequent and conditionally dependent sub-actions. The intuition is that when an EV needs to be repositioned, one could firstly decide which region it should be redirected to, and then subsequently determine which station within that region should be its new destination. Therefore, the expan-

[†]Work done during an internship at the University of Warwick.
[*]Corresponding author.

sion dynamics are localized within individual regions, while the first sub-actions have static action spaces. In particular, the proposed action cascading approach uses two connected policy networks to generate the sub-actions in sequel.

There is also a solid body of existing work that uses the MARL formulation in rebalancing shared mobility systems. For instance, the recent work in [Li *et al.*, 2018] uses a spatial-temporal DQN to rebalance the shared bikes, but it is fundamentally different from this paper since it doesn't consider the dynamic system expansion. On the other hand, the recent fleet management work [Li *et al.*, 2019; Lin *et al.*, 2018; Zhou *et al.*, 2019] tackles the order dispatching problem in ride sharing, which although different from our problem, share similar challenges in the varying action spaces. However, their solution is to allow the agents to directly rank the potential actions (selecting local orders) and choose the one with the highest score, while we use two policy networks to generate cascading actions which first localize and then handle the non-stationarity. Concretely, the contributions of this paper are as follows:

- To the best of our knowledge, we are the first to identify the problem of rebalancing expanding EV sharing systems. We formulate the incentive-based rebalancing problem with the MARL framework, and design the agents, states and rewards for the EV context accordingly.
- We propose a novel approach of policy optimization with action cascading, which uses two connected policy networks to handle dynamics introduced by rapid expansion of the EV sharing systems. We also design a regularized reward which can effectively stabilize training.
- We build a simulator which is calibrated with 12 months' operation data collected from a real-world EV sharing system*. The proposed approach has been evaluated extensively, and results show that it significantly outperforms the state-of-the-art, offering up to 12% improvement in net revenue and 14% in demand satisfied rate.

2 Problem Statement

In this section, we first introduce some key concepts and assumptions of the EV sharing systems considered in this paper, highlighting their unique properties. Then we describe the problem of incentive-based EV rebalancing in the presence of continuous system expansion.

Electric Vehicles (EVs). We assume that the EVs used in our system are of limited range. During normal driving the remaining range can be determined by a typical discharging model, while the charging time is estimated by a charging model given battery capacities and charger specifications [Tremblay and Dessaint, 2009]. In Sec. 4 we will show how different EV ranges and charging duration may impact the patterns of system operation in more detail.

EV Sharing Stations. The EV sharing systems considered in this paper are *station-based*, i.e. users only rent or return EVs from/to the online stations, where the EVs shall be charged. We represent a station s as a tuple $(\texttt{loc}, \#\texttt{c}, \#\texttt{v})$, where $\texttt{loc}$ is the geographic coordinates (e.g. latitude and longitude), $\#\texttt{c}$ is the total number of charging docks, and $\#\texttt{v}$ is the number of EVs initially equipped in the station. We assume when a station s was newly deployed for operation, it had $\#\texttt{v}$ EVs available to rent and $\#\texttt{c}$ - $\#\texttt{v}$ free spaces for vehicle returns ($\#\texttt{v} < \#\texttt{c}$).

System Expansion. Unlike most of the existing work, we assume the EV sharing system is continuously evolving during its operation. At any discrete timestamp t, new EV stations could be deployed in new areas to extend coverage, or within already covered areas to increase density. On the other hand, stations can also be closed for various reasons, e.g. limited profit. We assume that overall the system keeps expanding, i.e., there are more stations being deployed than closed.

Incentive-based Rebalancing. Let $o_t = (s^{\text{o}}, s^{\text{d}})$ be an order placed by a user at time t, requesting to rent an EV from station s^{o} and return to s^{d}. To alleviate the imbalanced EV distribution within the system, we may have to reposition the EV serving this order o_t to another station $s^{\text{d}'}$ instead of the original destination s^{d}, if the remaining range is sufficient. We motivate the user who is driving the EV to perform this for us by offering a monetary reward of value $d(s^{\text{d}}, s^{\text{d}'})$, which depends on the extra distance she has to drive from s^{d} to $s^{\text{d}'}$. The user may or may not accept this offer according to a prior user model [Singla *et al.*, 2015]. If she accepts, we pay the reward directly e.g., discounting the order price, while otherwise we allow the user to return the EV to her original destination s^{d} and charge the order normally.

Therefore, the rebalancing problem studied in this paper is that given the total available budget B on user incentives, for each order $o_t = (s^{\text{o}}, s^{\text{d}})$, we want to decide where to reposition the EV to minimize the future customer loss (i.e. satisfying as much user demand as possible) while maximizing the net revenue of the EV sharing system, in the presence of limited EV range, typical EV charging time, and the dynamically expanding station network.

3 Methodology

In this section, we first formulate the EV rebalancing problem as a Multi-Agent Reinforcement Learning (MARL) task with non-stationary action spaces (Sec. 3.1). Then we present the proposed policy optimization approach with action cascading in Sec. 3.2, which is able to handle such non-stationarity.

3.1 EV Rebalancing as a MARL Task

We model the rebalancing problem in EV sharing systems as a Markov Game $G = (N, \mathcal{X}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where at most N agents interact with the environment, characterized by the states $\mathcal{X}$ and transition function $\mathcal{T}$. $\mathcal{A}$ is the joint actions of the agents, $\mathcal{R}$ is the reward function, and γ is the discount factor. Concretely, they are defined as follows:

Agents. We assume the space is partitioned into hexagonal grids, each of which is managed by an agent, i.e., controls the rebalancing operations among EV stations within a region. As the EV sharing system is continuously expanding, the population of agents at a time t is a variable N_t, but we assume the maximum number of agents are fixed N, which is the maximum possible hexagonal grids in the space partition.

States. At time t, the global state $\boldsymbol{x}_t$ is the combination of states for each hexagonal grid $\boldsymbol{x}_t = \{x_t^i\}$, $i \in [1, N]$. For the i-th grid, x_t^i encodes information about the stations within

*Code available at https://github.com/ev-sharing/simulator.

this grid. In particular, for each station we consider its location #loc, number of available charging docks #c, number of EVs parked in the stations #v and their individual range, as well as the potential future rent/return requests and the mean value of future orders in the next timestamp.

Agent Observations. We assume an agent can make observations of state $\boldsymbol{x}_t$ from grids within its two-hop neighborhood, i.e. it can observe states of itself x_t^i and the 19 grids around it. This enables agents to interact with the local environment and learn to cooperate with their neighboring agents.

Actions. For agent i, its action a_t^i describes how each EV returned to the grid i at time t will be repositioned (there should be multiple EVs returned to i). We assume our agents only reposition the EVs to stations within one-hop neighborhood to avoid excessive user effort. In our system stations can be deployed or closed dynamically, therefore the action space $\mathcal{A}_t^i$ of agent i is non-stationary, i.e., the reposition candidates may vary over time.

State Transitions. The state transition probabilities $\mathcal{T}$ are defined as $\mathcal{T}(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{a}_t, \boldsymbol{u}_t)$, where $\boldsymbol{x}_t$ is the previous state, $\boldsymbol{a}_t$ is the joint action, and $\boldsymbol{u}_t$ is the system dynamics, capturing the station network expansion, i.e., which new stations are deployed with how many new EVs, and which existing stations are made off-line from t.

Reward Function. In the rebalancing task we would like to maximize the revenue of our EV sharing system with minimum cost on user incentives. Intuitively, revenue can be increased by sending more EVs to stations with higher order values. However, in practice we found that this would lead to greedy agents that only push EVs to certain "hot" stations such as airports but ignore the others, causing further imbalance. Therefore, to mitigate that we also reward the agents that reposition EVs to stations in shortage of vehicles. Concretely, to balance fairness and the potential revenue, we design reward function r_t^i as:

$$r_t^i = g_t^{\mathrm{d}'} + \alpha_1 v_t^{\mathrm{d}'} + \alpha_2 b_t^{\mathrm{d}'} - \alpha_3 d(s^{\mathrm{d}'}, s^{\mathrm{d}}) \tag{1}$$

where $g_t^{\mathrm{d}'}$ is the expected demand gap at the reposition candidate $s^{\mathrm{d}'}$ in the next timestamp, i.e., number of orders minus number of available EVs onsite, and $v_t^{\mathrm{d}'}$ is the expected order value at $s^{\mathrm{d}'}$. $b_t^{\mathrm{d}'}$ indicates if $s^{\mathrm{d}'}$ is empty, which explicitly encourages agents to reposition EVs to the currently empty stations. The penalty term $d(s^{\mathrm{d}'}, s^{\mathrm{d}})$ is the cost (monetary reward) we pay, which is proportional to the squared extra distance to $s^{\mathrm{d}'}$ [Pan *et al.*, 2019]. The weights α_1, α_2 and α_3 scale different reward/penalty terms to approximately the same range, which are determined empirically via grid search. Given the reward function, each agent i aims to maximize its discounted reward $\mathbb{E}[\sum_{k=0}^{\infty} \gamma^k r_{t+k}^i]$.

3.2 Policy Optimization with Action Cascading

In our MARL formulation, the action spaces of the agents are non-stationary, due to the fact that the EV station network is dynamically evolving over time. We now present the proposed policy optimization approach with action cascading (ac-PPO), which extends the standard algorithms to handle such non-stationarity. The key intuition is that the action of repositioning an EV to an alternative station can be viewed as a sequence of two sub-actions, where we first decide which grid the EV should go to, and then figure out which station within that selected grid should be the new destination. In essence, we chain two sub-actions, one *inter-grid* and the other *intra-grid*, where the former can have fixed action spaces, and the non-stationarity of the station network would only affect the latter. In the following, we first explain the design of action cascading in more detail, and then we show how we adapt the reward structure to stabilize training.

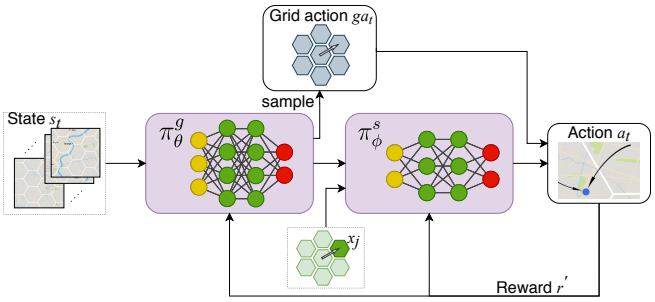


Figure 1: Overview of the proposed action cascading.

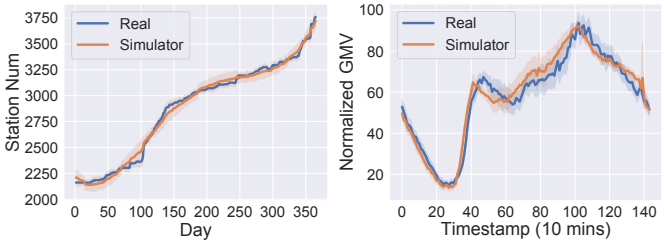
Action Cascading. Let a_t^i be the action of agent i at time t. For simplicity here we assume there is only one EV returned to grid i at t. For the cases with multple EVs, we could simply handle them in batches. We decomposite a_t^i as $a_t^i = (ga_t^i, sa_t^i)$, where ga_t^i is the inter-grid action that decides which grid within the neighborhood the EV should be redirected to, and sa_t^i is the intra-grid action which determines the actual destination station within the selected grid. Here ga_t^i has a fixed action space, which contains the six neighbors around the grid i and itself. Therefore, ga_t^i can be sampled from the output of a standard policy network π_θ^g. Assume that we have a ga_t^i that would redirect the EV to a nearby grid j. Now we need to find the intra-grid action sa_t^i that selects a suitable station within grid j. Note that here the action space of sa_t^i is not stationary, as there are always stations deployed or closed in grid j. We address this by using an action-in policy network π_ϕ^s as shown in Fig. 1, which takes the current state x_t^j of the grid j, and the output of the last layer of the inter-grid policy network π_θ^g as input. The former contains information of all current stations and vehicles within the target grid j, while the latter can be viewed as the context which encodes the determined inter-grid action ga_t^i. The output of the network π_ϕ^s are scores of each station within grid j, and we select the one with highest value as the desired action sa_t^i.

Essentially, we use two policy networks that are connected, to determine the inter-grid and intra-grid actions respectively. We train the networks with the clipped objective function:

$$L^{\mathrm{CLIP}}(\theta, \phi) = \hat{\mathbb{E}}\left[\min(R_t^\theta \hat{A}_t^{\theta,\phi}, \mathrm{Clip}(R_t^\theta, 1-\epsilon, 1+\epsilon)\hat{A}_t^{\theta,\phi}\right] \tag{2}$$

where R_t^θ is the probability ratio between new and old inter-grid policy: $R_t^\theta = \frac{\pi_\theta^g(\boldsymbol{ga}_t|\boldsymbol{x}_t)}{\pi_{\theta_{\mathrm{old}}}^g(\boldsymbol{ga}_t|\boldsymbol{x}_t)}$. ϵ is a hyperparameter (usually set to 0.2$\sim$0.3). On the other hand, the advantage function $\hat{A}_t^{\theta,\phi}$ considers both inter and intra-grid policies, since the reward $\boldsymbol{r}_t$ is given to the full actions $\boldsymbol{a}_t = (\boldsymbol{ga}_t, \boldsymbol{sa}_t)$.

Reward Regularization. Essentially, the proposed ac-PPO



(a) (b)

Figure 2: Simulator calibration. (a) Simulated vs. real station network expansion for one year (averaged over 10 runs), and (b) Simulated vs. real gross Merchandise Value (GMV).

addresses the non-stationarity in action spaces by decomposing the action into the sequence of inter-grid and intra-grid sub-actions, where we fit the problem into the policy optimization framework by allowing non-stationary reward. In fact, the reward distribution of the same action across different timestamps may vary, since the set of candidate reposition stations are changing, leading to large gradient variance when training π_θ^g. To address that, we propose to regularize the reward function r_t^i in Eq. (1) with a baseline: $r_t^{i'} = r_t^i + \beta \bar{r}_t(j)$. Here $\bar{r}_t(j) = \bar{v}_t(j) \cdot \bar{g}_t(j)$ is the product of the mean order value $\bar{v}_t(j)$ and the average future demand gap $\bar{g}_t(j)$ (# of user demand - # of available EVs) per station in grid j, assuming that the action is to reposition the EV to a station in the target grid j. Intuitively, $\bar{r}_t(j)$ can be viewed as the "potential" of the grid, indicating how much extra revenue one would expect to get if more EVs are repositioned to this grid. The weight β scales the regularization term to adjust its impact during learning.

4 Evaluation

4.1 Simulator Design

To support training and evaluation of the proposed MARL algorithm, we design a simulator which is calibrated with real-world data from an EV sharing system.

Simulator Settings. In our simulator, we consider 10 mins as one timestamp, i.e., one day contains 144 intervals. The space is partitioned into hexagonal grids, where in total we have 598 grids covering the entire city. We use a random process to simulate the dynamic system expansion, i.e., where to deploy new stations, and which existing stations should be closed. To simulate the user demand, we train a neural network which takes the current station network as input, and generates demand for all online stations. The simulator assumes all the EVs in the system are fully charged at initialization, and operate with a known charging/discharging model.

When there is a need to reposition an EV, the simulator computes a monetary reward depending on the square of the extra distance that the user has to travel [Singla *et al.*, 2015]. We assume the user would accept with a probability p, which depends on the distance already traveled and the extra distance to cover. If it is accepted, the simulator updates the status of order, the EV and the new destination accordingly.

Simulator Calibration. We calibrate our simulator with real EV sharing data, collected in Shanghai for 12 months. The data includes both order records and the expansion process of the station network, i.e., when and where a station was deployed or closed. We also collected key meta-data, including station locations, numbers of charging docks in each station, the charging/discharging models of the EVs, etc. As shown in Fig. 2(a), the patterns of simulated system expansion are very close to the actual expansion during the year, with Pearson correlation 0.9957 and $p <$ 1e-10. For demand generation, we use the calibrated system expansion, and tune the simulator with respect to the Gross Merchandise Value (GMV). Fig. 2(b) shows that the simulated order data has very similar properties in GMV with the real data, with Pearson correlation 0.9599 and $p <$ 1e-10.

4.2 Experimental Settings

We compare the proposed approach with the following baselines:

- No Rebalancing (NR), which simulates the operation of EV sharing system without any rebalancing actions.

- Random Rebalancing (RND), where EVs are repositioned randomly to nearby stations.

- Revenue Greedy (REV), which is similar with RND but selects the stations with the highest average order values.

- Demand Gap Greedy (DMD), which prefers the stations with the highest demand gap in the vicinity.

- STRL, which is our implementation of the the state-of-the-art approach [Li *et al.*, 2018]. It uses multi-agent spatial-temporal reinforcement learning to reposition shared bikes.

We also consider different variants the *inter-grid* policy network π_θ^g as follows:

- Policy Gradient (ac-PG), which uses the standard policy gradient technique to determine the inter-grid actions.

- Deep Q Networks (ac-DQN), which uses a deep Q-network to approximate the action-state values.

- Advantage Actor Critic (ac-A2C), which uses actor and critic networks to determine actions and estimate the advantage values respectively.

- Proximal Policy Optimization (ac-PPO), which is the proposed policy optimization approach as discussed in Sec. 3.2.

For all the above variants, we use our intra-grid policy network π_ϕ^s as described in Sec.3.2 and the same reward function. In addition, we consider different approaches to determine the *intra-grid* actions, while fixing π_θ^g as PPO:

- PPO + Random (PPO+RND), which randomly selects a destination station within the grid determined by PPO.

- PPO + Revenue Greedy (PPO+REV), which finds the station with the highest average order value as destination.

- PPO + Demand Gap Greedy (PPO+DMD), which selects destination station as the one with the largest demand gap.

All the competing approaches are implemented with TensorFlow 1.14.0, and trained with a single NVIDIA 2080Ti GPU. We evaluate them against two main metrics: i) **Demand Satisfied Rate (DS)**, which is the percentage of the demand satisfied by an algorithm w.r.t total user demand; and ii) **Net Revenue Value (NV)**, which is calculated as the GMV subtracts the cost on user incentives.

4.3 Results

Overall Rebalancing Performance. The first set of experiments evaluates the overall rebalancing performance of dif-

	NR	RND	REV	DMD	STRL	ac-DQN	ac-PG	ac-A2C	ac-PPO	PPO+RND	PPO+REV	PPO+DMD
DS	74.69%	49.79%	82.15%	81.09%	82.47%	83.50%	83.64%	85.23%	**88.79%**	53.94%	83.19%	82.88%
ΔDS	—	-24.90%	7.46%	6.41%	7.78%	8.81%	8.95%	10.55%	**14.10%**	-20.75%	8.51%	8.20%
ΔGMV	—	-36.30%	8.25%	3.22%	9.27%	10.76%	11.26%	13.13%	**18.13%**	-29.82%	10.41%	6.22%
ΔNV	—	-47.71%	-7.64%	-0.48%	1.12%	6.95%	7.37%	8.53%	**12.23%**	-48.40%	-3.27%	1.72%
$\Delta\|\boldsymbol{o}\|/\|\boldsymbol{a}\|$	–	—	9.28	4.98	2.08	1.14	1.09	1.11	1.12	—	7.82	3.53

Table 1: Performance of the competing approaches in 1) demand satisfied rate (DS), 2) increased demand satisfied rate (ΔDS) w.r.t. baseline NR, 3) increased % of GMV (ΔGMV) w.r.t. baseline NR, 4) increased % of net revenue value (ΔNV) w.r.t. baseline NR, and 5) # of increased order per reposition operation ($\Delta|\boldsymbol{o}|/|\boldsymbol{a}|$, only showing positive values).

ferent approaches, as shown in Table. 1. We allow the station network to expand at the normal speed. We see that RND won't help at all while if we are greedy on order values (REV) we do satisfy more demand by 7%, but the net revenue drops by 8%. This is because with this algorithm, the agents tend to excessively reposition EVs to stations with high order values, while ignoring the cost on user incentives. We find that on average, REV would satisfy one extra order at the cost of repositioning 9.3 EVs. On the other hand, DMD achieves more balanced performance, improving the demand satisfied rate (DS) by 3%, while maintaining similar net revenue with the baseline NR. The state-of-the-art STRL outperforms the baselines, with 8% improvement in DS and 1% improvement in NV: on average it repositions 2.1 EVs to satisfy an extra order. It confirms that by using spatial-temporal RL, the STRL can better learn the demand pattern across space and time. However, we see that our approaches significantly outperforms STRL. For instance, ac-PPO can achieve almost 15% improvement in DS, while obtaining 12% more NV. In addition we find that ac-PPO only needs to reposition 1.1 EVs to satisfy an extra order, which is very efficient.

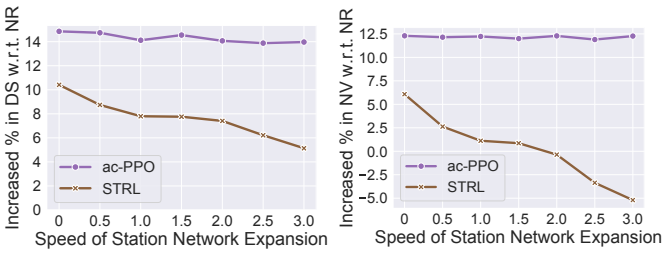
Performance of Inter-grid Policy. This experiment compares the performance of different algorithms in learning the inter-grid policy π_θ^g in our action cascading framework. We only vary π_θ^g while using the same intra-grid policy network π_ϕ^s later and feed the algorithms with the same reward. We see that even the weakest performed algorithm ac-DQN can achieve better performance than STRL by 5% improvement in DS and 1% in NV, since STRL doesn't have the mechanism of handling station network expansion. On the other hand, policy gradient (ac-PG) only performs slightly better than ac-DQN, but is inferior to ac-A2C. The best ac-PPO (see Sec. 3.2) provides a further improvement of 4% in both DS and NV, achieving 14.10% better DS and 12.23% NV than no rebalancing (NR). This projects to approximately 200,000 USD extra revenue per month according to the real data where the mean order value is 3.8 USD and average number of orders per month is about 500k.

Performance of Intra-grid Policy. The third set of experiments studies the performance of different ways to determine the intra-grid actions. We compare ac-PPO with three variants, where we replace the intra-grid policy network π_ϕ^s with different rule-based strategies. We see that random approach (PPO-RND) produces worse results than baseline NR. The PPO-REV is more sensible but as discussed above, it tends to perform lots of unnecessary repositions, causing undesirable performance in NV. We observe similar trend in PPO-DMD, which offers similar DS (8% improvement) and slightly better NV (2% improvement). As expected, the proposed ac-PPO performs the best overall, and the gap between ac-PPO and PPO-DMD is about 10% in NV and 6% in DS. This confirms that the two sub-actions should be optimized jointly, and the proposed π_ϕ^s outperforms the rule-based baselines.

Impact of System Expansion Dynamics. This set of experiments investigates the impact of system expansion dynamics to rebalancing algorithms. Here we only consider the state-of-the-art STRL and the proposed ac-PPO. We adjust the simulator to allow different speeds of expansion, i.e., on average how many new stations should be deployed and existing stations closed per day. We vary the speed from 0 to 3, where 0 means the station network is static, and 3 means station network expands at 3x speed comparing to that in the real world. As shown in Fig. 3, we see that when there is no dynamics at all, the gap between STRL and ac-PPO is only about 4% in DS, and 6% in NV. However, as the system begins to expand, the performance of STRL drops immediately. At the normal speed the gap between STRL and our ac-PPO is 6% in DS and 11% in NV. In the extreme case where the expansion speed is 3x, the gap in DS becomes almost 10%, while STRL can't increase the NV when the expansion speed is above 1.5. This is expected as STRL relies heavily on station clustering which would fail when the system expands, leading to inferior decisions in rebalancing. On the other hand, we see that the ac-PPO approach is very robust as the expansion speed increases, confirming that the proposed action cascading can work well under different levels of expansion dynamics.

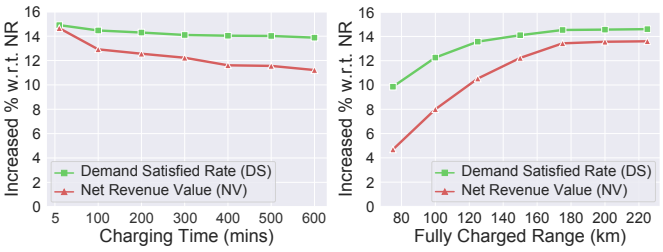
Performance vs. Charging Time. In this experiment, we study a practical problem in the EV sharing industry: how charging time would affect the rebalancing performance. Here we fix the EV range at 150km, and vary the charging time from 5min to 600min. The 5min lower bound corresponds to battery swapping used in some EVs (e.g. bit.ly/NIOPower). Fig. 4(a) shows the DS and NV increased by ac-PPO w.r.t baseline NR at different charging speeds. Clearly as the charging time increases, the performance gain drops. This is expected because we can't perform any reposition action when an EV is being charged. However, if the EVs are battery replaceable, the increase in NV is about 3% comparing to the standard case with 300min charging time. This indicates the approach of battery swapping does have its merits and should be considered in practice. On the other hand, the gap between the fastest and slowest charging is negligible in DS, and about 3% in NV. This means our ac-PPO is very robust to different charging speeds: even systems with slow chargers could enjoy considerable performance boost.

Performance vs. Battery Capacity. The last set of exper-



(a) Increased % in DS (b) Increased % in NV

Figure 3: Performance of the proposed ac-PPO with STRL under different speeds of station network expansion.



(a) (b)

Figure 4: Performance of the proposed ac-PPO algorithm vs. (a) charging time, and (b) EV fully charged range.

iments studies the impact of battery capacity to rebalancing performance. This indicates how EV sharing systems using different EV models (short range vs. long range) would behave under rebalancing strategies. Here we fix the charging time at 300min and vary the EV range from 75km to 225km. Fig. 4(b) shows the increased DS and NV of ac-PPO compared to the baseline NV. We see that as the range increases, the performance gain becomes more significant. This makes sense because EVs with longer range require less frequent charging, and often allow more flexible rebalancing: they could be repositioned to further stations if needed. We also observe that the performance is more sensitive for EVs with shorter range, e.g. the performance gain in NV is halved with 75km range. However, even in that case our ac-PPO offers 10% improvement in DS and 5% more NV, which is still better than state-of-the-art. On the other hand, we see after the range increases over 175km, the benefit becomes negligible.

5 Related Work

Shared Mobility System. Recently, shared mobility systems have attracted extensive interests from various communities [Jiang *et al.*, 2018; Xu *et al.*, 2018; Furuhata *et al.*, 2013; Dillahunt *et al.*, 2017]. Comparing to traditional systems [Li *et al.*, 2018; Singla *et al.*, 2015], systems with EVs are more complex due to their unique properties, such as range limitations and long charging time. A solid body of work has looked into various new problems and challenges in this context, such as route planning and optimization [Sarker *et al.*, 2018; Yuen *et al.*, 2019], charging scheduling [Yan *et al.*, 2018; Yuan *et al.*, 2019; Wang *et al.*, 2019a], and infrastructure planning [Sarker *et al.*, 2018; Du *et al.*, 2018]. Our work complements the existing studies which primarily consider electric taxis or buses, in that we focus on the EV sharing systems which operate in a very different way. In addition, unlike existing work which often assumes the system is static, we investigate the rebalancing problem in the context of continuous system expansion.

Rebalancing Shared Mobility Services. Existing work to address the problem of rebalancing shared mobility services can be broadly categorized into three types, static reposition [Liu *et al.*, 2016; Raviv *et al.*, 2013], dynamic reposition [Ghosh *et al.*, 2016a; Singla *et al.*, 2015; Ghosh *et al.*, 2017; Li *et al.*, 2018; Wang *et al.*, 2019b; Etienne and Latifa, 2014] and user-based reposition. The first two are conducted by the system operators while the last is performed by users. This paper falls into the last category, which solves the rebalancing problem by incentivizing the users with rewards to rent or return vehicles at designated stations [Singla *et al.*, 2015; Pan *et al.*, 2019]. However, unlike existing solutions which assume the system is static, we aim to tackle the rebalancing problem in the presence of dynamically changing station networks. This is fundamentally different from the static cases as at different time the candidates for reposition operations may be different, which can't be addressed by the existing rebalancing approaches.

Deep Reinforcement Learning in Mobility. Due to their distributed nature, many mobility applications such as traffic control, fleet management and rebalancing [Li *et al.*, 2019; Lin *et al.*, 2018; Li *et al.*, 2018; Pan *et al.*, 2019] can be modeled as multi-agent games, which can be well solved by deep reinforcement learning. For instance, the work in [Li *et al.*, 2019] addresses the order dispatching problem for ride sharing systems using mean field MARL, while [Lin *et al.*, 2018] proposes a contextual MARL framework to tackle the fleet management problem. Another work in [Li *et al.*, 2018] considers a spatio-temporal reinforcement learning approach, to dynamically reposition shared bikes across different areas. In this paper, we also model the rebalancing problem in EV sharing system with MARL framework. However, our work differs from the existing work in that a) we extend the existing framework to directly model unique properties of EV sharing such as range limitations and charging time, and more importantly b) we develop the new action cascading technique to support continuous system expansion.

6 Conclusion

In this paper, we study the incentive-based rebalancing for continuous expanding EV sharing systems. We formulate the rebalancing task as a MARL problem, and solve it using the proposed policy optimization with action cascading. We design a simulator to simulate the operation of EV sharing systems, which is calibrated with real data from an actual EV sharing system for a year. Extensive experiments have shown that the proposed approach significantly outperforms the baselines and state-of-the-art in both satisfied demand rate and net revenue, and is robust to different levels of system expansion dynamics. We also show that the proposed approach performs consistently with different charging time and EV range. For future work, we would like to explore the more realistic case where the system can be rebalanced by both incentivized users and the dedicated staff, while there are heterogeneous EV models in operation.

References

[Dillahunt *et al.*, 2017] Tawanna R Dillahunt, Xinyi Wang, Earnest Wheeler, Hao Fei Cheng, Brent Hecht, and Haiyi Zhu. The sharing economy in computing: A systematic literature review. *CSCW*, 1:38, 2017.

[Du *et al.*, 2018] Bowen Du, Yongxin Tong, Zimu Zhou, Qian Tao, and Wenjun Zhou. Demand-aware charger planning for electric vehicle sharing. In *KDD*, pages 1330–1338. ACM, 2018.

[Etienne and Latifa, 2014] Côme Etienne and Oukhellou Latifa. Model-based count series clustering for bike sharing system usage mining: a case study with the vélib'system of paris. *ACN TIST*, 5(3):39, 2014.

[Furuhata *et al.*, 2013] Masabumi Furuhata, Maged Dessouky, Fernando Ordóñez, Marc-Etienne Brunet, Xiaoqing Wang, and Sven Koenig. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57:28–46, 2013.

[Ghosh *et al.*, 2016a] Supriyo Ghosh, Michael Trick, and Pradeep Varakantham. Robust repositioning to counter unpredictable demand in bike sharing systems. In *IJCAI*. AAAI Press, 2016.

[Ghosh *et al.*, 2016b] Supriyo Ghosh, Jing YuKoh, and Patrick Jaillet. Improving customer satisfaction in bike sharing systems through dynamic repositioning. In *IJCAI*. AAAI Press, 2016.

[Ghosh *et al.*, 2017] Supriyo Ghosh, Pradeep Varakantham, Yossiri Adulyasak, and Patrick Jaillet. Dynamic repositioning to reduce lost demand in bike sharing systems. *Journal of AI Research*, 58:387–430, 2017.

[Jiang *et al.*, 2018] Shan Jiang, Le Chen, Alan Mislove, and Christo Wilson. On ridesharing competition and accessibility: Evidence from uber, lyft, and taxi. In *WWW*, pages 863–872, 2018.

[Kooti *et al.*, 2017] Farshad Kooti, Mihajlo Grbovic, Luca Maria Aiello, Nemanja Djuric, Vladan Radosavljevic, and Kristina Lerman. Analyzing uber's ride-sharing economy. In *WWW*, pages 574–582, 2017.

[Li *et al.*, 2018] Yexin Li, Yu Zheng, and Qiang Yang. Dynamic bike reposition: A spatio-temporal reinforcement learning approach. In *KDD*, pages 1724–1733. ACM, 2018.

[Li *et al.*, 2019] Minne Li, Zhiwei Qin, Yan Jiao, Yaodong Yang, Jun Wang, Chenxi Wang, Guobin Wu, and Jieping Ye. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *WWW*, pages 983–994. ACM, 2019.

[Lin *et al.*, 2018] Kaixiang Lin, Renyu Zhao, Zhe Xu, and Jiayu Zhou. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In *KDD*, pages 1774–1783. ACM, 2018.

[Liu *et al.*, 2016] Junming Liu, Leilei Sun, Weiwei Chen, and Hui Xiong. Rebalancing bike sharing systems: A multi-source data smart optimization. In *KDD*, pages 1005–1014. ACM, 2016.

[Pan *et al.*, 2019] Ling Pan, Qingpeng Cai, Zhixuan Fang, Pingzhong Tang, and Longbo Huang. A deep reinforcement learning framework for rebalancing dockless bike sharing systems. In *AAAI*, pages 1393–1400, 2019.

[Raviv *et al.*, 2013] Tal Raviv, Michal Tzur, and Iris A Forma. Static repositioning in a bike-sharing system: models and solution approaches. *EURO Journal on Transportation and Logistics*, 2(3):187–229, 2013.

[Sarker *et al.*, 2018] Ankur Sarker, Haiying Shen, and John A. Stankovic. Morp: Data-driven multi-objective route planning and optimization for electric vehicles. *IMWUT*, 1(4):162:1–162:35, January 2018.

[Shaheen *et al.*, 2018] Susan Shaheen, Adam Cohen, and Mark Jaffee. Innovative mobility: Carsharing outlook. 2018.

[Singla *et al.*, 2015] Adish Singla, Marco Santoni, Gábor Bartók, Pratik Mukerji, Moritz Meenen, and Andreas Krause. Incentivizing users for balancing bike sharing systems. In *AAAI*, 2015.

[Tremblay and Dessaint, 2009] Olivier Tremblay and Louis-A Dessaint. Experimental validation of a battery dynamic model for ev applications. *World electric vehicle journal*, 3(2):289–298, 2009.

[Wang *et al.*, 2019a] Guang Wang, W. Li, J. Zhang, Y. Ge, Z. Fu, F. Zhang, Y. Wang, and Desheng Zhang. shared-charging: Data-driven shared charging for large-scale heterogeneous electric vehicles. *IMWUT*, May 2019.

[Wang *et al.*, 2019b] Shuai Wang, Tian He, Desheng Zhang, Yunhuai Liu, and Sang H Son. Towards efficient sharing: A usage balancing mechanism for bike sharing systems. In *WWW*, pages 2011–2021. ACM, 2019.

[Wei *et al.*, 2017] Chong Wei, Yinhu Wang, Xuedong Yan, and Chunfu Shao. Look-ahead insertion policy for a shared-taxi system based on reinforcement learning. *IEEE Access*, 6:5716–5726, 2017.

[Xu *et al.*, 2018] Zhe Xu, Zhixin Li, Qingwen Guan, Dingshui Zhang, Qiang Li, Junxiao Nan, Chunyang Liu, Wei Bian, and Jieping Ye. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *KDD*, pages 905–913. ACM, 2018.

[Yan *et al.*, 2018] Li Yan, Haiying Shen, Zhuozhao Li, Ankur Sarker, John A. Stankovic, Chenxi Qiu, Juanjuan Zhao, and Chengzhong Xu. Employing opportunistic charging for electric taxicabs to reduce idle time. *IMWUT*, 2(1):47:1–47:25, March 2018.

[Yuan *et al.*, 2019] Yukun Yuan, Desheng Zhang, Fei Miao, Jiming Chen, Tian He, and Shan Lin. p^2 charging: Proactive partial charging for electric taxi systems. In *ICDCS*, 2019.

[Yuen *et al.*, 2019] Chak Fai Yuen, Abhishek Pratap Singh, Sagar Goyal, Sayan Ranu, and Amitabha Bagchi. Beyond shortest paths: Route recommendations for ride-sharing. In *WWW*, pages 2258–2269. ACM, 2019.

[Zhou *et al.*, 2019] Ming Zhou, Jiarui Jin, Weinan Zhang, Zhiwei Qin, Yan Jiao, Chenxi Wang, Guobin Wu, Yong Yu, and Jieping Ye. Multi-agent reinforcement learning for order-dispatching via order-vehicle distribution matching. In *CIKM*, pages 2645–2653. ACM, 2019.