Deep Reinforcement Learning for Real-time Dynamic Aggregation and Scheduling of Electric Vehicles

Mingyang Zhang, *Student Member*, Hongrong Yang, *Student Member, IEEE*, Yinliang Xu, *Senior Member, IEEE*, and Hongbin Sun, *Fellow IEEE*

***Abstract*—It is recognized that large-scale electric vehicles (EVs) can be aggregated and behave as a controllable storage to provide flexibility for power systems. To provide high-quality services to both the system and EV users, it is critical to accurately estimate the aggregated flexibility of EVs, which is highly challenging due to the uncertainties from regulation signals and EV behaviors. Thus, this paper proposes a learning-based approach, which is online, model-free and adaptive. First, the optimal aggregated flexibility offering problem is formulated as a Markov decision process (MDP). Moreover, a heuristic causal real-time scheduling policy is developed to allocate the disaggregated power to each EV. Then, a deep reinforcement learning (DRL) algorithm, Munchausen Soft Actor-Critic (M-SAC), is proposed to solve the MDP problem. M-SAC algorithm contains an unsupervised learning stage for state-dimension reduction and a reinforcement learning stage for the decision-making, accelerated by Munchausen method. Numerical simulation results based on the real-world dataset demonstrate that the proposed approach can effectively deal with multiple uncertainties and balance economy and user satisfaction, and its performance is superior to existing model-driven aggregation methods and DRL algorithms.**

***Index Terms*-- Electric vehicle, online aggregation, scheduling policy, deep reinforcement learning, uncertainty, soft actor-critic.**

I. Introduction

The current world energy structure is experiencing great transitions to tackle the energy crisis. Renewable energy sources (RESs) and electric vehicles (EVs) are under rapid deployment to decarbonize the electricity and transportation sectors, respectively [1].

Power systems require more flexibility to cope with the integration of large-scale RESs. However, it is costly to simply depend on the supply side. With the support of internet of things (IoT) technology, EVs can respond to dispatch signals, which not only reduces the adverse impact of unregulated charging, but also provides low-cost flexibility services to power systems [2].

Direct control of a large-scale EVs is not realistic for the system operator, therefore, the formulation of an aggregation model is necessary to unlock the flexibility of EVs [3]. The current literature in EV aggregation can be categorized in two main groups. The first group is to capture the aggregated feasible region geometrically [4]-[7]. Boundary aggregation is a common-used method that obtains the approximate aggregated flexibility by linearly summing up the energy and power bounds [4], [5]. Polytope-based methods utilize polyhedral set containment principle to compute the approximate Minkowski sum of the polytopes describing the feasible region of each EV [6]. Fourier-Motzkin elimination is applied in [7] to derive the exact formulation of aggregate feasible region. The second group is to model the dynamics of EVs population [8]-[10]. Wang [8] and Kiani [9] develop a state-space model (SSM) of aggregated EVs and estimate the upward and downward flexibility capacities from it. Wu [10] consider real-time updating of SSM parameters to enhance aggregation accuracy.

While the above works provides fruitful techniques for quantifying EVs aggregated flexibility in an *offline* manner, these methods assume that the aggregator has non-causal information about the exact realization or stochastic distribution of future events. However, in practice, the behaviors and energy states of EVs can hardly be estimated precisely in advance.

Therefore, facing the great uncertainties in EVs' behaviors, *online* aggregation strategies based solely on the up-to-date information are more promising paradigm. A box-type inner-approximate online aggregation method based on model predictive control (MPC) is proposed in [11] to schedule the base power and flexibility reserve. However, the MPC-based methods still rely on forecasts and are computationally demanding due to the rolling optimization process. An online Lyapunov optimization-based aggregation algorithm is developed in [12] to characterize the EVs' flexibility region. All above-mentioned aggregation methods are *model-driven*, which have the following limitations: 1) The approximate results are often either overly optimistic or conservative; meanwhile, some aggregation algorithms are computationally burdensome and difficult to apply in real time. 2) There are two types of EV charging uncertainties: sudden arrival and sudden departure [13]. The existing methods are not effective for the latter one.

Alternatively, *learning-based* approach is a promising candidate for decision-making under uncertain scenarios because it can directly learn the optimal strategy from experience data without the information of randomness distribution. Model-free deep reinforcement learning (DRL) frameworks have been utilized to handle the charging scheduling problem in the literature. For example, constrained policy optimization and the soft actor-critic (SAC) framework are adopted in [14], [15] and [16] to find the optimal charging/discharging strategy for an individual EV, respectively. In contrast, Refs. [17]-[20] use DRL to learn a collective charging plan for EV populations in an EV charging station (EVCS). To maximize the profit of the EVCS, Refs. [17] and [18] develop online scheduling control algorithms based on SARSA and actor-critic learning, respectively. To flatten the load profile, Refs. [19] and [20] modify the proximal policy

optimization (PPO) and Q-learning algorithms to learn the optimal charging policy. However, as the number of aggregated EVs increases, the size of the problem's solution space will also grow dramatically, leading to an unacceptably long training time [21]. Therefore, the dimensionality reduction techniques are required when dealing with large-scale EVs.

While the charging scheduling has been intensively investigated, the online learning-based EVs flexibility aggregation has not been explored, which is the focus of this paper. Compared to the charging problem, the online flexibility aggregation is more complicated: there is a coupling between the base power and aggregated flexibility, on top of that, it is difficult to figure out their future impact since one does not know which charging rate will be chosen within the flexibility interval. This means that the aggregator can obtain higher financial benefits through strategic bidding, meanwhile facing a greater risk of penalty for not being able to satisfy the charging demand.

The allocation of disaggregated power to each EV is another important task of the aggregator. The optimization-based disaggregation method is adopted in [11] and [18] with the objective of minimizing the discrepancy. This method is computationally demanding and more suitable for day-ahead applications. For the real-time allocation, Earliest Deadline First (EDF) and Least Laxity First (LLF) are two widely used sorting-based heuristic scheduling policies [22]. However, only the charging case is considered in these policies [23].

To fill the aforementioned gaps, this paper proposes a DRL-based online EV aggregation method to obtain the optimal energy and flexibility bidding strategy under multiple uncertainties. Firstly, the aggregation problem is formulated as a Markov decision process (MDP), which is then solved by the proposed model-free DRL algorithm. Furthermore, a heuristic causal real-time scheduling policy is developed to allocate the disaggregated power. The main contributions are summarized as follows:

1) To the best of authors' knowledge, this study for the first time addresses the problem of online aggregation of EVs flexibility using a DRL approach. Compared with the existing model-driven methods, the proposed approach can offer a more economical aggregated flexibility and can better deal with uncertainties from the regulation signal and EV charging behaviors.

2) A real-time scheduling policy, named Improved Less Laxity and Longer Remaining Processing Time, is proposed for power allocation. Compared with the existing scheduling policies, the proposed policy works for both charging and discharging cases, and improves the charging fulfillment by assigning the base power.

3) From the solution algorithm aspect, this study combines the SAC algorithm with Munchausen reinforcement learning to form the Munchausen-SAC framework. Furthermore, this framework integrates an auto-encoder for state-dimension reduction. Compared with the traditional DRL methods, the proposed algorithm shows advantages in terms of learning speed and the final performance.

II. Problem Statement

A. Operational framework

A distribution network is chosen as the platform for flexibility reporting of EVCS in this paper. The distribution system operator (DSO) is considered as a balancing entity responsible for maintaining the safe and reliable operation of the distribution system. To economically mitigate RESs uncertainties in real-time operation, DSO will incentivize flexible load aggregators such as EVCS to report their aggregated flexibility intervals. This paper focus on the efficient aggregation of EVs using a DRL approach to earn revenue and minimize total cost by providing flexibility reserves to the distribution system.

B. Real-time dynamic aggregation and scheduling

In this paper, an online dynamic aggregation and scheduling strategy is developed, as shown in Fig. 1, which can be operated in real-time based only on the information of the EVs that have already arrived at the charging station. The proposed EVCS-DSO interaction framework is as follows, which contains two EVCS decision phases:

i. The DSO broadcasts the electricity price c_t and reward coefficients for upward/downward flexibility reserve c_t^u/c_t^d for the time step *t*.
ii. First decision phase: based on the broadcasted prices, combined with information about all plugged-in EVs, EVCS reports its aggregate base charging power p_t and flexibility interval $[\overline{p}_t, \underline{p}_t]$ to the DSO.
iii. According to the collected flexibility intervals and the output of RESs generation, the DSO performs real-time dispatch and determines the regulation signal $p_t^{\mathrm{Reg}} \in [\overline{p}_t, \underline{p}_t]$ for the EVCS.
iv. Second decision phase: after receiving the reference signal, the EVCS disaggregates it into individual charging tasks according to a certain algorithm, namely the real-time scheduling policy $\vartheta(p_t^{\mathrm{Reg}})$, and performs charging decisions on plugged-in EVs.
v. Move to the next time step and repeat from step i) with the updated information.

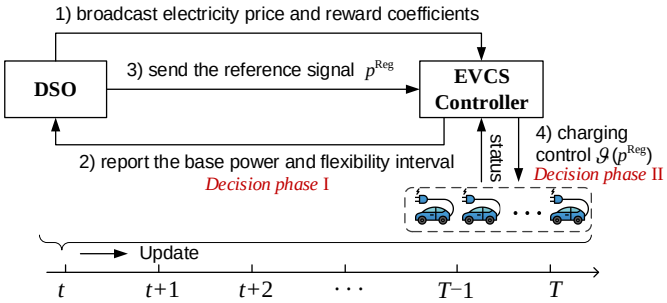


Fig. 1. Schematic of online dynamic aggregation and scheduling framework.

Obviously, the above aggregated energy and flexibility bidding problem is a multi-period optimization problem, where the decisions should be made sequentially with the uncertain information being revealed gradually over time. And every current decision may affect future decisions as well as total returns. However, the decision at time *t* would depend on the

information available up to time *t*, but not on the results of future observations. In this way, this problem can be constructed as a multi-stage stochastic dynamic programming:

$$\max_{\mathbf{A}\mathbf{x}_1 \le h(\mathbf{b}_1)} f_1(\mathbf{x}_1) + \mathbb{E}^{\boldsymbol{\xi}_2}\left[\max_{\mathbf{A}\mathbf{x}_2 \le h(\mathbf{b}_2, \mathbf{B}_2\mathbf{x}_1)} f_2(\mathbf{x}_2, \boldsymbol{\xi}_2) + \mathbb{E}^{\boldsymbol{\xi}_3|\boldsymbol{\xi}_2}\left[\cdots + \mathbb{E}^{\boldsymbol{\xi}_T|\boldsymbol{\xi}_{T-1}}\left[\max_{\mathbf{A}\mathbf{x}_T \le h(\mathbf{b}_T, \mathbf{B}_T\mathbf{x}_{T-1})} f_T(\mathbf{x}_T, \boldsymbol{\xi}_T)\right]\right]\right] \quad (1)$$

$$f_t(\mathbf{x}_t, \boldsymbol{\xi}_t) := \left[-c_t p_t^{\text{Reg}} + c_t^u\left(p_t - \underline{p}_t\right) + c_t^d\left(\bar{p}_t - p_t\right)\right]\Delta t \quad (2)$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \mathbf{B}_t = \begin{bmatrix} 0 & B_t & 1-B_t \end{bmatrix}, B_t \in [0,1] \quad (3)$$

s.t.:

$$\left[u_{1.t}, u_{2,t}, ..., u_{N_t,t}\right] = \vartheta\left(p_t^{\text{Reg}}\right), \forall t \quad (4)$$

$$\underline{p}_t \le p_t^{\text{Reg}} = \mathbf{B}_{t+1}\mathbf{x}_t \le \bar{p}_t \quad (5)$$

$$-\bar{u}^d \le u_{i,t} \le \bar{u}^c, \forall t, \forall i \quad (6)$$

$$\sum_{t=t_i^a}^{\tilde{t}_i^d} \max(u_{i,t}, 0)\Delta t \eta_c + \min(u_{i,t}, 0)\Delta t / \eta_d = d_i^{req}, \forall i \quad (7)$$

where $\mathbf{x}_t = [p_t, \bar{p}_t, \underline{p}_t]^T$ is the decision vector. $\xi_1 := [c_1, c_1^u, c_1^d, \mathbf{b}_1]$ is nonrandom observed data vector, $\xi_t := [c_t, c_t^u, c_t^d, \mathbf{B}_t, \mathbf{b}_t]$, t=2,…,T, are random data vectors. $\mathbf{A}$ and $\mathbf{B}_t$ are parameter matrices shown in (3). $\mathbf{b}_t$ is a vector about the state of the EVCS, reflecting the uncertainty of charging demand and EV connections, and $\mathbf{B}_{t+1}\mathbf{x}_t$ denotes the reference signal given by the DSO to the EVCS for time t; (2) represents the profit of the EVCS at time t. $h()$ is a function that calculates the maximum power range that the EVCS can currently provide, and constraint $\mathbf{A}_t\mathbf{x}_t \le h(\mathbf{b}_t, \mathbf{B}_t\mathbf{x}_{t-1})$ is used to limit the reported aggregated flexibility interval should be within feasible range. ϑ denotes the scheduling policy for charging control of each EV. $u_{i,t}$ is the charging rate of EV i in time t. we assume that all charging bays in the charging station have the same maximum charging rate $\bar{u}^c$ and discharging rate $\bar{u}^d$. t_i^a and $\tilde{t}_i^d$ denote the arrival and departure time of EV i, respectively, and the latter one is uncertain. d_i^{req} is the charging demand of EV i. η_c and η_d are the charging and discharging efficiency of the devices, respectively. Δt is the interval between two adjacent time slots.

However, finding the transition probabilities of the randomness ξ_t is not a trivial task, and it is difficult to solve this optimization problem using an intuitive mathematical programming approach. Therefore, a model-free DRL algorithm is proposed in the following sections.

III. System Model, MDP and Scheduling Policy

In this section, firstly, the dispatch model of distribution system is introduced. Then, a heuristic causal scheduling policy is proposed for the real-time allocation of the disaggregated charging power. Finally, the EV aggregation problem is formulated as a Markov Decision Processes (MDP).

A. Dispatch model of the DSO

1) Set-point problem: OPF. At the beginning of time t, the DSO solves the optimal power flow (OPF) model based on the aggregated charging power p_t reported by the EVCS, as well as the latest forecast information of RESs:

$$\min_{\mathbf{x}_1^{DSO}} \sum_k f_{g,k}\left(P_{G,k,t}^{ref}\Delta t\right) + \pi_t^{EM} P_{PCC,t}^{ref}\Delta t \quad (8)$$

s.t.:

$$\underline{P}_{G,k} \le P_{G,k,t}^{ref} \le \bar{P}_{G,k} \quad (9)$$

$$\underline{Q}_{G,k} \le Q_{G,k,t}^{ref} \le \bar{Q}_{G,k} \quad (10)$$

$$\mathbf{v} = \mathbf{D}[\mathbf{x}_1^{DSO} \quad \mathbf{w}^f \quad \mathbf{p}]^T + \mathbf{d} \quad (11)$$

$$\mathbf{i}_L = \mathbf{E}[\mathbf{x}_1^{DSO} \quad \mathbf{w}^f \quad \mathbf{p}]^T + \mathbf{e} \quad (12)$$

$$\underline{\mathbf{v}} \le \mathbf{v} \le \bar{\mathbf{v}}, \ \underline{\mathbf{i}}_L \le \mathbf{v} \le \bar{\mathbf{i}}_L \quad (13)$$

where the objective is to minimize the generation cost and the power purchasing cost from the wholesale market. π_t^{EM} is the energy market price. The vector of decision variables is $\mathbf{x}_1^{DSO} = \{P_{G,k,t}^{ref}, Q_{G,k,t}^{ref}, P_{PCC,t}^{ref}, Q_{PCC,t}^{ref}\}$. $P_{PCC,t}^{ref}$ and $P_{G,k,t}^{ref}$ donate the scheduled power injection at point of common coupling (PCC) point and the output of generator k. $w_{m,t}^f$ is the forecast output of RES m. $\bar{P}_{G,k}/\underline{P}_{G,k}$ and $\bar{Q}_{G,k}/\underline{Q}_{G,k}$ denote the maximum and minimum active power and reactive power output of generator k, respectively. Eqs. (11)-(12) are linear power flow model and (13) is network constraint. $\mathbf{v}$ and $\mathbf{i}_L$ denote the vector of nodal voltage and line current magnitudes, respectively, and $\bar{\mathbf{v}}/\underline{\mathbf{v}}$ and $\bar{\mathbf{i}}_L/\underline{\mathbf{i}}_L$ are their upper/lower limits. Matrices $\mathbf{D}$, $\mathbf{E}$ and vectors $\mathbf{d}$, $\mathbf{e}$ are system all parameters, the details can refer to [11].

2) Corrective control problem: Uncertainty Mitigation. In the dispatch interval, the actual output of RES $w_{m,t}^{real}$ may deviate from the prediction. The corrective control seeks effective corrective actions of non-renewable units and flexible load in response to the prediction error. The following optimization problem is used to produce the corrective actions of obligated generators and the EVCS:

$$\min_{\mathbf{x}_2^{DSO}} \sum_k c_{G,k}\left(P_{G,k,t} - P_{G,k,t}^{ref}\right)^2 + c_P\left(P_{PCC,t} - P_{PCC,t}^{ref}\right)^2 \quad (14)$$

s.t.:

$$\underline{P}_{G,k} \le P_{G,k,t} \le \bar{P}_{G,k} \quad (15)$$

$$\underline{Q}_{G,k} \le Q_{G,k,t} \le \bar{Q}_{G,k} \quad (16)$$

$$\underline{p}_t \le p_t^{\text{Reg}} \le \bar{p}_t \quad (17)$$

$$\mathbf{v} = \mathbf{D}[\mathbf{x}_1^{DSO} \quad \mathbf{w}^{real} \quad \mathbf{p}^{\text{Reg}}]^T + \mathbf{d} \quad (18)$$

$$\mathbf{i}_L = \mathbf{E}[\mathbf{x}_1^{DSO} \quad \mathbf{w}^{real} \quad \mathbf{p}^{\text{Reg}}]^T + \mathbf{e} \quad (19)$$

where the objective function is to minimize the adjustment cost of generators while penalizing the power deviations at point of PCC, and $c_{G,k}$ and c_P are their cost coefficients. The vector of decision variables is $\mathbf{x}_2^{DSO} = \{P_{G,k,t}, Q_{G,k,t}, P_{PCC,t}, Q_{PCC,t}, p_t^{\text{Reg}}\}$. $P_{G,k,t}$ and $P_{PCC,t}$ are the output of generator k and power injection at PCC point after adjustment, respectively. Eq. (13) is also included in the constraints.

B. Real-time scheduling policy (Decision phase II)

After receiving the information of p_t^{Reg}, EVCS needs to disaggregate it to each charging task according to a certain policy. A popular policy, LLF, based on the charging laxity, which is defined as follows for the charging task at time t:

$$\phi_{i,t} = t_{i,t}^{rem} - t_{i,t}^{\min} \quad (20)$$

where *i* denotes the index of charging bay, and we also use *i* to refer to the charging task or EV that is currently occupying charging bay *i*. $t_{i,t}^{min} = d_{i,t}^{req}/(\overline{u}^c \eta_c \Delta t)$ is the minimal time period needed to finish the charging task *i*. $d_{i,t}^{req}$ and $t_{i,t}^{rem}$ denote the residual energy demand and time slots left till departure of the charging task *i*, respectively. Charging laxity can be seen as a measure of the degree to which a charging task can be deferred; the greater the laxity of the task, the greater the flexibility it can provide.

The Laxity and Longer Remaining Processing Time (LLLP) algorithm proposed in [23] is a modified version of LLF that also compares the processing time of tasks, i.e., $t_{i,t}^{min}$, on the basis of laxity:

***Definition 1*:** For any two charging task *i* and *j*, we say $i \prec j$ at time *t* (*j* has higher priority in the charging order over *i*) if we have: i) $\phi_{i,t} > \phi_{j,t}$ or ii) $\phi_{i,t} = \phi_{j,t}$ and $t_{i,t}^{min} < t_{j,t}^{min}$.

However, LLLP can only handle charging cases and has a rate of zero or a fixed number (i.e., simple on-off control). In this paper, we propose an Improved LLLP (ILLLP) algorithm, which is adaptive for both charging and discharging cases with variable charging rates. Before proceeding with the proposed scheduling policy, we first calculate the charging power boundaries of task *i* at time *t* as follows:

$$u_{i,t}^{\min} = \max\left\{-\overline{u}^d, (d_{i,t}^{req} - d_{i,t}^{\max})\eta_d / \Delta t\right\} \tag{21}$$

$$u_{i,t}^{Base} = \begin{cases} \min\left\{\overline{u}^c, \left(1-\phi_{i,t}\right)\overline{u}^c\right\}, & \phi_{i,t} < 1 \\ 0, & \phi_{i,t} \geq 1 \end{cases} \tag{22}$$

$$u_{i,t}^{\max} = \min\left\{\overline{u}^c, d_{i,t}^{req}/\eta_c \Delta t\right\} \tag{23}$$

where $u_{i,t}^{min}$ and $u_{i,t}^{max}$ represent the minimum and maximum charging power at time *t*, respectively. $d_{i,t}^{max}$ is maximum energy demand of task *i*, which equals the target energy value minus the minimum allowable battery energy value. $u_{i,t}^{Base}$ is the base charging power given to task *i* to ensure its charging completion for the remaining time.

A pseudocode of the proposed ILLLP is presented in Algorithm 1 and illustrated as follows:

Step 1) Sort all active charging tasks T_i ($i \in \mathcal{N}_t$) according to Definition 1 in the order of priority from the highest to the lowest, denoted as T^1, T^2, ..., $T^{|\mathcal{N}_t|}$. $\mathcal{N}_t = \left\{i \middle| t_{i,t}^{rem} > 0\right\}$ is the set of all charging bays that have EV plug-in at time *t*.

Step 2) To ensure charging completion, $u_{i,t}^{Base}$ is first allocated to all charging tasks. Then, the remaining aggregate power g_t can be calculated.

Step 3) g_t is further allocated to the tasks according to their priorities. Two cases are discussed here, namely, charging and discharging cases.

i) If g_t is greater than zero, it would be allocated to the first task T^1 in the sorting table. For the power in excess of the maximum power $u_{1,t}^{max}$ of T^1, it would be allocated to the second task T^2. This process continues until g_t is expended, or all active tasks are serviced at their maximum values. If g_t is still greater than 0 after this process is finished, g_t will be recorded as the surplus power g_t^s.

ii) If g_t is less than zero, reduce the power of the charging task to the minimum value $u_{i,t}^{min}$ in order from T^{N_t} to T^1 in the table until g_t is equal to zero or all active tasks are serviced at $u_{i,t}^{min}$. If g_t is still less than 0 after this process is finished, g_t will be recorded as the deficit power g_t^d.

Algorithm 1 ILLLP scheduling policy.

Input: p_t^{Reg} for the EVCS and the state $\mathbf{s}_t$ at time *t*.
Output: Charging decisions $u_{i,t}$ for each plugged-in EVs.

1: Sort all active tasks T_i, $i \in \mathcal{N}_t$, in order of priority from highest to lowest to form a sorted table.
2. Allocate $u_{i,t} = u_{i,t}^{Base}$ to all tasks in the table and the remaining aggregate power is given as $g_t = p_t^{\text{Reg}} - \sum_{i \in \mathcal{N}_t} u_{i,t}$.
3. **If** $g_t \geq 0$, **then**
4. **for** *i*=1: 1: $|\mathcal{N}_t|$ **do**
5. Increase $u_{i,t}$ until $u_{i,t} = u_{i,t}^{max}$ or the updated $g_t = 0$. **If** $g_t = 0$, **break**.
6. **end for**
7. **If** the updated $g_t > 0$, let $g_t^s = g_t$. **end if**
8. **else**
9. **for** *i*=$|\mathcal{N}_t|$: -1: 1 **do**
10. Decrease $u_{i,t}$ until $u_{i,t} = u_{i,t}^{min}$ or the updated $g_t = 0$. **If** $g_t = 0$, **break**.
11. **end for**
12. **If** the updated $g_t < 0$, let $g_t^d = -g_t$. **end if**
13. **end if**

C. MDP formulation (Decision phase I)

We consider the EVCS operator as an intelligent agent and the physical system as the environment. The agent needs to make sequential decisions in the uncertain environment and maximize its cumulative reward. Mathematically, the decision-making problem can be formulated as a MDP as follows:

1) State: The state is the input of the agent. Here, the state at time *t* is defined as:

$$\mathbf{s}_t \triangleq \left[t, C_t, S_{i,t}^{bay}\middle|_{\forall i \in \mathcal{N}}\right] \tag{24}$$

where *t* is the current time period, which reflects the EV arrivals, total charging demand and the electricity prices in the near future. $C_t = [c_t, c_t^u, c_t^d]$ is the vector of observed real-time price and reward rates. $S_{i,t}^{bay} = [d_{i,t}^{req}, d_{i,t}^{max}, t_{i,t}^{rem}]$ is the state vector of the charging bay *i*. $\mathcal{N} := \{1,2,\ldots,N\}$, *N* is the total number of charging bays at the charging station. It is worth noting that not all charging bays are occupied at every time step, for occupied charging bays, $t_{i,t}^{rem} > 0$, and for idle charging bays, $t_{i,t}^{rem} = 0$.

2) Action: Based on $\boldsymbol{s}_t$, the EVCS decides the base charging power and flexibility interval. To adapt the MDP model to the proposed DRL method, we set the action vector as $\boldsymbol{a}_t = [p_t, \alpha_t^{up}, \alpha_t^{dn}]$. Due to the limitation of charging infrastructures, the action should satisfy:

$$-|\mathcal{N}_t|\overline{u}^d \leq p_t \leq |\mathcal{N}_t|\overline{u}^c \tag{25}$$

$$0 \leq \alpha_t^{up} \leq 1,\ 0 \leq \alpha_t^{dn} \leq 1 \tag{26}$$

$$\overline{p}_t = p_t + \alpha_t^{up}\left(|\mathcal{N}_t|\overline{u}^c - p_t\right) = |\mathcal{N}_t|\alpha_t^{up}\overline{u}^c + \left(1-\alpha_t^{up}\right)p_t \tag{27}$$

$$\underline{p}_t = p_t - \alpha_t^{dn}\left(|\mathcal{N}_t|\overline{u}^d + p_t\right) = -|\mathcal{N}_t|\alpha_t^{dn}\overline{u}^d + \left(1-\alpha_t^{dn}\right)p_t \tag{28}$$

where α_t^{up} and α_t^{dn} represent the percentage of flexibility downward/upward reserve to the maximum downward/upward power margin, respectively. The flexibility interval can be expressed as (27)-(28) using α_t^{up} and α_t^{dn}.

3) State transition: The transition probability $\mathcal{P}(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})$ is influenced by many factors, such as regulation signal of the DSO, the arrival and leave of EVs, etc. To be more practical, we consider $\mathcal{P}$ to be unknown. After receiving the regulation signal, the EVCS allocate it to each charging task following the proposed policy ILLLP. Under the charging schedule $[u_{i,t}|_{\forall i\in\mathcal{N}_t}] = \vartheta(p_t^{\text{Reg}})$, task state are updated as:

$$d_{i,t+1}^{req} = \begin{cases} d_{i,t}^{req} - u_{i,t}\Delta t\eta_c, & u_{i,t} \geq 0 \\ d_{i,t}^{req} - u_{i,t}\Delta t/\eta_d, & u_{i,t} < 0 \end{cases} \tag{29}$$

$$t_{i,t+1}^{rem} = \begin{cases} t_{i,t}^{rem} - 1, & t_{i,t}^{rem}\Delta t \geq \delta \\ \mathcal{S}(t_{i,t}^{rem})(t_{i,t}^{rem} - 1), & t_{i,t}^{rem}\Delta t < \delta \end{cases} \tag{30}$$

Here, we consider (30) to stimulate the sudden departure of EVs, where δ is a threshold of remaining parking time, below which the EV has a certain probability of departure at the next time period. $\mathcal{S}()$ is a function of $t_{i,t}^{rem}$ with an output of 0 or 1. The smaller $t_{i,t}^{rem}$ is, the higher the probability that $\mathcal{S}()$ outputs 0. Only after $t_{i,t}^{rem} = 0$, a newly arrived EV will be connected to this charging bay and then $S_{i,t}^{bay}$ will become profiles of this EV.

4) Reward: The object is to maximize the profit of the EVCS, which is the difference between the revenue of providing flexibility and the energy procurement cost. Meanwhile, EVCS has to realize accurate tracking of the regulation signal and guarantee the customer satisfaction. Thus, the reward function is designed to contain four parts as:

$$r_t(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1}) = r_{1,t} + r_{2,t} + r_{3,t} + r_{4,t} \tag{31}$$

$$r_{1,t} = \omega_1\left[-c_t p_t^{\text{Reg}} + c_t^u(p_t - \underline{p}_t) + c_t^d(\overline{p}_t - p_t)\right]\Delta t \tag{32}$$

$$r_{2,t} = -\omega_2\left(g_t^s + g_t^d\right)\Delta t \tag{33}$$

$$g_t^s = \max\{0, p_t^{\text{Reg}} - \sum_{i\in\mathcal{N}_t} u_{i,t}^{\max}\} \tag{34}$$

$$g_t^d = \max\{0, \sum_{i\in\mathcal{N}_t} u_{i,t}^{\min} - p_t^{\text{Reg}}\} \tag{35}$$

$$r_{3,t} = -\omega_3 \sum_{i:t_{i,t}^{rem}\neq 0, t_{i,t+1}^{rem}=0} u(d_{i,t+1}^{req}) \tag{36}$$

$$r_{4,t} = \omega_4 \sum_{i\in\mathcal{N}_t} \min\{0, \phi_{i,t}\} \tag{37}$$

where ω_1 to ω_4 are the positive weight factors. Eq. (32) is the profits of EVCS and (33) is the penalty for failure to complete the regulation signal. The surplus power g_t^s and deficit power g_t^d can be obtained by (35). Eq. (36) denotes the dissatisfaction function of customers to penalize unfulfilled energy at the time of departure. Because $r_{3,t}$ is only manifested at the departure times of EVs, it is difficult for the agent to learn effective experience during the exploration period. Therefore, $r_{4,t}$ is introduced to guide the algorithm to converge towards the goal. This design makes the reward denser and yields better training results.

The aim of the aggregate energy and flexibility bidding problem is to seek the optimal policy π which maximizes the expected discounted rewards in an operational horizon:

$$\max_{\pi} J = \mathop{\mathbb{E}}_{\tau\sim\pi}\left[\sum_{t=0}^{T}\gamma^t \cdot r_t\right] \tag{38}$$

where π is the policy that generates action according to state. $\tau = \{\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{a}_{T-1}, \boldsymbol{s}_T\}$ is a trajectory.

IV. Deep Reinforcement Learning Algorithm

In this section, we first introduce the continuous SAC algorithm briefly. Then, our proposed DRL algorithm is presented to solve the above MDP.

A. Soft Actor-Critic Algorithm

In the SAC framework, the objective is augmented with an entropy term to balance the exploration and exploitation:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{(\boldsymbol{s}_t,\boldsymbol{a}_t)\sim\rho_\pi}\left[\sum_{t=0}^{T}\gamma^t\left[r_t(\boldsymbol{s}_t,\boldsymbol{a}_t) + \alpha\mathcal{H}\left(\pi\left(\cdot|\boldsymbol{s}_t\right)\right)\right]\right] \tag{39}$$

where ρ_π denotes the state-action marginal of the trajectory distribution induced by $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$. α is the temperature coefficient that governs the relative importance of the entropy to the reward. $\mathcal{H}(\pi(\cdot|s))$ is the entropy that controls the randomness of the optimal policy and is calculated as $\mathcal{H}\left(\pi(\cdot|s)\right) = -\log(\pi(\cdot|s))$.

The SAC algorithm has two deep neural networks (DNNs): the Q network parameterized by $\theta = \{\theta_1, \theta_2\}$ and the policy network parameterized by ϕ. Using SAC to train the models has two main steps: policy evaluation and policy improvement.

1) policy evaluation. We need learn the soft Q-function:

$$Q(\boldsymbol{s}_t,\boldsymbol{a}_t) = r_t(\boldsymbol{s}_t,\boldsymbol{a}_t) + \gamma\mathbb{E}_{\boldsymbol{s}_{t+1}\sim\mathcal{P}}V(\boldsymbol{s}_{t+1}) \tag{40}$$

where $V(\,)$ is the soft state value function and is given by:

$$V(\boldsymbol{s}_t) = \mathbb{E}_{\boldsymbol{a}_t\sim\pi}\left[Q(\boldsymbol{s}_t,\boldsymbol{a}_t) - \alpha\log\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)\right] \tag{41}$$

The SAC introduces a DNN to approximate the soft Q-function as $Q_\theta(\boldsymbol{s}_t,\boldsymbol{a}_t)$. And the parameters can be trained by minimizing the soft Bellman residual by utilizing previously sampled state and action stored in the replay buffer $\mathcal{D}$:

$$J_Q(\theta) = \mathbb{E}_{(\boldsymbol{s}_t,\boldsymbol{a}_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(\boldsymbol{s}_t,\boldsymbol{a}_t) - \left(r_t(\boldsymbol{s}_t,\boldsymbol{a}_t) + \gamma\mathbb{E}_{\boldsymbol{s}_{t+1}\sim\mathcal{P}}\left[V_{\bar{\theta}}(\boldsymbol{s}_{t+1})\right]\right)\right)^2\right] \tag{42}$$

where $V_{\bar{\theta}}(\boldsymbol{s}_{t+1})$ is the estimated soft state value using a target network, and its parameters can be updated by a moving average method.

2) policy improvement. Since the aggregated charging power of the EVCS is continuous, the policy function π_ϕ is defined as a Gaussian distribution: $\pi_\phi(\boldsymbol{a}_t|\boldsymbol{s}_t)\sim\mathcal{N}(\mu,\sigma^2)$, which is also approximated using a DNN. Then, the parameters of the policy network can be learned by minimizing the expected Kullback-Leibler divergence:

$$J_\pi(\phi) = \mathbb{E}_{\boldsymbol{s}_t\sim D}\left[\mathbb{E}_{\boldsymbol{a}_t\sim\pi_\phi}\left[\alpha\log\left(\pi_\phi(\boldsymbol{a}_t|\boldsymbol{s}_t)\right) - Q_\theta(\boldsymbol{s}_t,\boldsymbol{a}_t)\right]\right] \tag{43}$$

Here, an automating entropy adjustment method to tune α is as follows:

$$J(\alpha)=\mathbb{E}_{\boldsymbol{a}_t\sim\pi_\phi}\left[-\alpha\log\pi_\phi\left(\boldsymbol{a}_t|\boldsymbol{s}_t\right)-\alpha\bar{\mathcal{H}}\right] \quad (44)$$

where $\bar{\mathcal{H}}$ is the desired minimum expected target entropy.

B. The Proposed DRL Algorithm

1) Auto-encoder based dimensionality reduction

It can be seen from (24) that as the number of charging bays increases, the dimension of state space will become very large, leading to the state-space explosion problem. In this paper, we integrate an unsupervised learning method, i.e., auto-encoder, into the RL framework for state dimension reduction.

An auto-encoder is a type of neural network trained to learn efficient representations of the input data and consists of two parts: an encoder and a decoder. The encoder and decoder can be considered as two functions implemented by neural networks: $\boldsymbol{z}=E_\psi(\boldsymbol{x})$, parametrized by ψ and $\boldsymbol{x}'=D_\omega(\boldsymbol{z})$, parametrized by ω. $E_\psi(\boldsymbol{x})$ maps $\boldsymbol{x}$ from data space to feature space, while $D_\omega(\boldsymbol{z})$ produces a reconstruction of the inputs.

In the state vector, $S_{i,t}^{bay}$ has three features, and to achieve dimensionality reduction, we aim to compress it into one feature. Therefore, the created auto-encoder is an *undercomplete auto-encoder* with input data $\boldsymbol{x}$ of 3-dimensions and codes $\boldsymbol{z}$ of 1-dimension. To facilitating training, we first normalize $S_{i,t}^{bay}$ to the numbers between 0~1 before inputting the samples. Given a dataset $\mathcal{B}=\{\boldsymbol{x}_i|\boldsymbol{x}_i\in\mathbb{R}^3,1\le i\le n\}$, the auto-encoder can be trained by minimizing a predefined loss function:

$$L(\psi,\omega)=\frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}_i-D_\omega\left(E_\psi\left(\boldsymbol{x}_i\right)\right)\right\|_2^2 \quad (45)$$

2) Munchausen-SAC

Munchausen RL (MRL) is a bootstrap method using the current policy, originally proposed in [24], and its core idea is to augment the scaled log-policy to the immediate reward when using any temporal difference scheme. In this paper, we combine the idea of MRL with SAC method to form the Munchausen-SAC (M-SAC) framework.

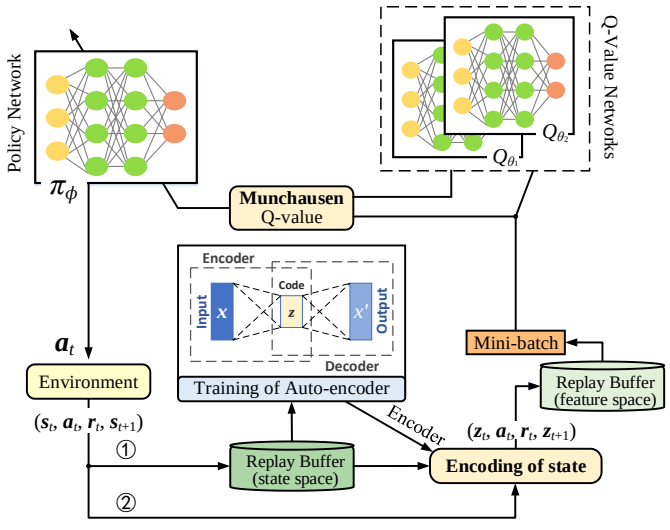


Fig. 2. The overall structure of the proposed DRL approach.

Traditional SAC algorithm is based on the maximum entropy reinforcement learning framework, which aims at maximizing both the expected return and entropy of the resulting policy, as shown in (39). In the proposed M-SAC framework, r_t is replaced by Munchausen reward by adding log-policy to immediate reward for optimization, so as to bootstrap the current agent's guess about what actions are good:

$$r_t^M\left(\boldsymbol{s}_t,\boldsymbol{a}_t\right)=r_t\left(\boldsymbol{s}_t,\boldsymbol{a}_t\right)+\eta\alpha\log\pi\left(\boldsymbol{a}_t|\boldsymbol{s}_t\right) \quad (46)$$

where $\eta\in[0,1]$ denotes the scaling factor. If the optimal policy π^* is known, the log-policy is positive for optimal actions and -∞ for sub-optimal actions. This is a strong signal to ease learning and direct the agent to suppress sub-optimal solutions, and adding it to the reward does not change the optimal policy.

Then, the soft Bellman equation and the update objective of the Q network in M-SAC framework are written as:

$$Q(\boldsymbol{s}_t,\boldsymbol{a}_t)=r_t^M\left(\boldsymbol{s}_t,\boldsymbol{a}_t\right)+\gamma\mathbb{E}_{\boldsymbol{s}_{t+1}\sim\mathcal{P}}V\left(\boldsymbol{s}_{t+1}\right) \quad (47)$$

$$J_Q(\theta)=\mathbb{E}_{(\boldsymbol{s}_t,\boldsymbol{a}_t)\sim D}\left[\frac{1}{2}\left(Q_\theta(\boldsymbol{s}_t,\boldsymbol{a}_t)-\left(r_t^M(\boldsymbol{s}_t,\boldsymbol{a}_t)+\gamma\mathbb{E}_{\boldsymbol{s}_{t+1}\sim\mathcal{P}}\left[V_{\bar{\theta}}(\boldsymbol{s}_{t+1})\right]\right)\right)^2\right] \quad (48)$$

Algorithm 2: Training Process of the Proposed DRL Algorithm.

Input: Initialized $\theta_1,\theta_2,\bar{\theta}_1,\bar{\theta}_2,\phi,\psi,\omega$; $\mathcal{D}\leftarrow\emptyset$.
Unsupervised Learning Stage:
Generate the training data set $\mathcal{B}$ for the auto-encoder (AE).
  for epoch = 1, 2, …, *K* **do**
    for mini-batch index= 1, 2, ... **do**
      Update the weights of AE neural networks using gradient in (45).
    end for
  end for
Derive the encoder $E_\psi(\cdot)$ (first half of the auto-encoder).
Reinforcement Learning Stage:
for episode=1, 2, …, *M* **do**
  Randomly select a day in the training dataset and obtain the initial state.
  for time step *t*=0, 1, …, *T* **do**
    Agent choose actions according to $\pi_\phi(\boldsymbol{a}_t|\boldsymbol{s}_t)$;
    Sample transition from the environment $\boldsymbol{s}_{t+1}\sim\mathcal{P}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)$;
    Apply the encoder $E_\psi(\cdot)$ to the transition$(\boldsymbol{s}_t,\boldsymbol{a}_t,r_t,\boldsymbol{s}_{t+1})$, transferring it into the feature space $(\boldsymbol{z}_t,\boldsymbol{a}_t,r_t,\boldsymbol{z}_{t+1})$ with $\boldsymbol{z}_t=E_\psi(\boldsymbol{s}_t)$.
    Store transition pair $(\boldsymbol{z}_t,\boldsymbol{a}_t,r_t,\boldsymbol{z}_{t+1})$ in replay buffer $\mathcal{D}$.
    If *t* is coming to the update step, **do**
      Sample mini-batch of experiences from $\mathcal{D}$;
      Calculate Munchausen reward in (46);
      Update critic network parameters using gradient in (48);
      Update policy network parameters using gradient in (43)
      Update temperature coefficient using gradient in (44);
      Update target critic network: $\bar{\theta}_i\leftarrow\tau\theta_i+(1-\tau)\bar{\theta}_i$.
    end if
  end for
end for
Output: $\theta_1,\theta_2,\phi,\psi,\omega$

3) Overall training framework

The overall structure of the proposed DRL approach is illustrated in Fig. 2, where two paths for encoding states are provided. The first one is to insert the training of the auto-encoder into the batch RL-framework with episodic exploration, which is more data-efficient but slows down the learning process. The second one is to train the auto-encoder separately and use it to extract the feature of state vector directly, which

has the opposite advantages and disadvantages of the first path. To better demonstrate the superiority of the proposed algorithm, the second path is taken in this paper.

Thereby, the training pipeline of the proposed approach consists of an unsupervised learning stage and a RL stage. The pseudocode of the final proposed DRL algorithm is summarized in Algorithm 2 and the details are explained in the following.

Firstly, the parameters of auto-encoder, policy and Q networks are initialized. In the unsupervised learning stage, we first build a dataset for S^{bay} using historical or simulation data. Then the parameters of the auto-encoder are optimized using batch gradient descent. Finally, the trained Encoder model is exported.

In the RL stage, the algorithm begins running with episodic iterations. At each time slot of each episode, the agent generates action based on the current policy. Correspondingly, the transition sample $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ is observed from the environments, which is then transferred into feature space using the encoder and stored in the replay buffer. Finally, a mini-batch of experiences is sampled to training the policy and Q networks using stochastic gradients in combination with MRL.

V. Case Study

In this section, we investigate the performance, convergence and scalability of the proposed method through multiple case studies using real-world data. All the simulations are executed in Python with PyTorch on a computer with an Intel Xeon E5-2690 CPU and 1 NVIDIA RTX 2080Ti GPU.

A. Simulation Setup

A small-scale system is selected here to verify the effectiveness of the proposed method. The test system consists of a modified IEEE 13-Node Test Feeder and an EVCS with 15 charging bays. EVs charging data is sourced from the UK transport apartment [25] in the range 2017.1.1-2017.12.31, which contains the data of each charging record such as charge-point identity, supplied energy, EV plug-in and plug-out times. We select first 20 days of each month for training, and the remaining days are used for performance evaluation, leading to 240/125 days in the training/test sets, respectively. The dynamic electricity price and flexibility reward rates are retrieved from PJM [26]. Similar to [20], we consider the scenario where the EVCS is located in the Central Business District (CBD), and most of the EVs have similar travel patterns, i.e., arriving at the parking lot in the morning and leaving in the afternoon. According to the statistical analysis in [15], the EV charging pattern in the office area has the distribution in Table I. Therefore, in the training set, we mix the simulated data with the real-world data to improve the generalization ability of the model. The hyper-parameters of the DRL algorithm are presented in Table II.

TABLE I. Distributions related to user's charging behavior.

	Distribution	Boundaries
Arrival time	N(9.2,1.5)	[6.5, 11.5]
Expected departure time	N(16.4,1.5)	[14, 19.5]
Required energy	N(15,5)	[2,30]

TABLE II. Hyper-parameters used in our experiments.

Parameters	Value
Optimizer	Adam
Number of hidden layers (All networks)	2
Number of hidden units per layer of policy network	256/64
Number of hidden units per layer of critic networks	256/32
Learning rate of actor network/critic network/temperature coefficient	0.0015/0.003/0.0015
Discount factor	0.99
Replay buffer size	1e6
Number of samples per mini batch	256
Nonlinearity	ReLU

B. Baseline Methods

We compare the performance of our proposed algorithm with the following baseline algorithms, including both learning-based and model-driven methods:

The proposed DRL algorithm is denoted by M-SAC. Learning-based baseline algorithms including three state-of-art DRL algorithm: 1) Traditional SAC; 2) Deep deterministic policy gradient (DDPG); 3) PPO.

Model-driven baseline algorithms including：

1) MPC-based traditional boundary aggregation [2], denoted by B1. This method utilizes the linear summation of energy and power boundaries of all individual EVs to represent the aggregate model. At each time step, the EVCS solves the optimal bidding problem based on the latest aggregate model (the prices for next T_p time steps are assumed to be known), but only declares the decisions for the first step.

2) The optimization-based aggregation method [11], which is also implemented in MPC framework, is denoted by B2. This approach provides a box-type inner approximate region of the feasible domain for the next T_p time steps of each EV, which ensures the feasibility of any regulation power trajectory in this flexibility interval.

Note that the learning-based methods need training process to optimize the parameters of neural networks before they are evaluated on the testing set. For the model-driven methods, they can be directly implemented on the testing set for evaluation.

C. Results and analysis

1) Training Performance

The evolution of the return, i.e., cumulative rewards, over training episodes of four DRL methods are depicted in Fig. 3. The solid curves correspond to the moving average and the shaded regions to the minimum and maximum episode return of every 20 episodes. The trace of the shaded region reflects the convergence speed and training stability of the algorithm.

To compare the performance of these methods, the neural networks of them are initialized with the same random seeds. As shown in Fig. 3, the agent's policy improves and the curve becomes more stable and finally converges as it interacts more with the environment. It can be found that the episode return during the training process and final converged return of the proposed method are higher than other methods. These results demonstrate the proposed approach has better training stability and exploration ability.

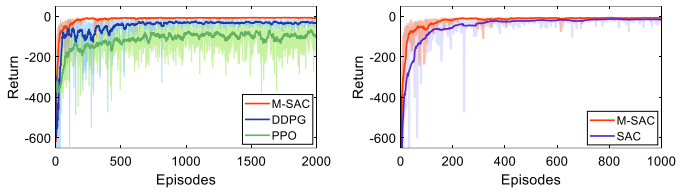


Fig. 3. Comparison of the training process of different algorithms.

Furthermore, the proposed M-SAC algorithm outperforms the traditional SAC algorithm in terms of convergence speed. In the training episodes 25 to 200, the average return change between two adjacent episodes for the MSAC and SAC are 0.31 and 0.18, respectively. This indicates that the return curve of M-SAC rises faster. This is because the update of the Q networks of M-SAC takes into account the effect of the current policy, which makes the gradient descent faster and smoother, and thus converges to the optimal solution more efficiently.

2) Profit and aggregation performance

The proposed algorithm is then evaluated on the test set. The cumulative total cost of the proposed and baseline methods over 125 test days are presented in Fig. 4. The percentage terms on the right axis denote the cost reduction ratio of the corresponding approach compared to the benchmark method, i.e., B1 method. And Table. III compares the average operational performance over all test days.

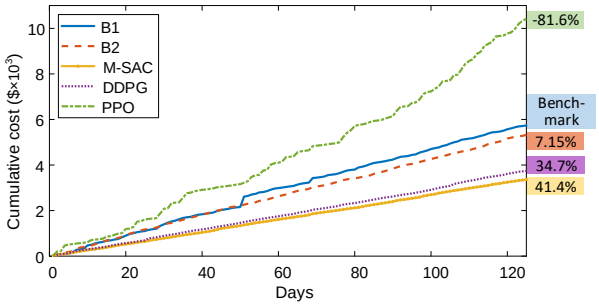


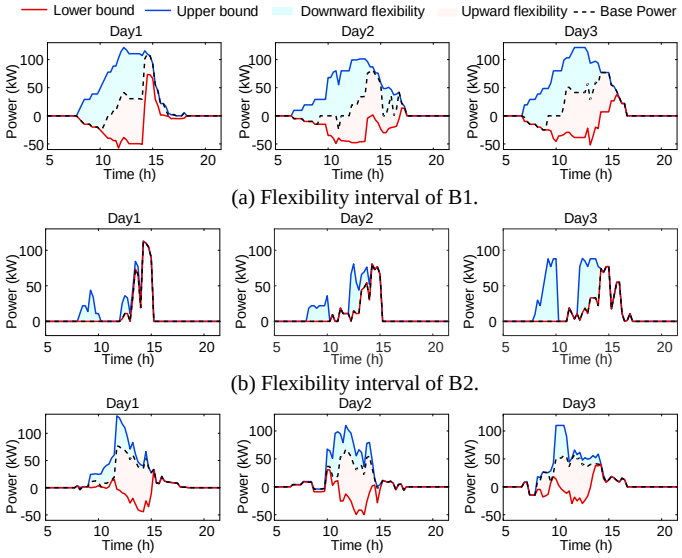
Fig. 4. Cumulative total cost on the test days.

Combining Fig. 4 and Table III, it can be observed that while B1 approach obtains the maximum flexibility reward, this outer approximation leads to an extension of the feasible region, while the uncertainty of EV departure time is not considered, which leads to lower charging completion and higher penalty cost. Similarly, B2 approach also fails to consider the departure time uncertainty, and still incurs a high penalty cost even if the flexibility interval offered is small. Among the three DRL methods, the proposed method performs the best, with a 41% reduction in total cost compared to the benchmark, the highest charging completion, and the smoothest rise of the curve, which indicates that the cost per day is basically the same and the strategy has a strong generalization ability; DDPG also performs well, but not as well as the proposed method in terms of the flexibility reward and charging completion; and PPO performs the worst, failing to find a satisfactory solution. It should be noted that because the departure time is random, it is impossible to completely satisfy the all the charging demand, and the proposed method is effective for learning a real-time strategy to minimize the total cost while adequately satisfying the user's charging demand.

TABLE III. Performance of various approaches on the test set.

	M-SAC	DDPG	PPO	B1	B2
Energy cost ($)	28.18	27.67	26.70	25.35	24.89
Flexibility reward ($)	4.82	2.78	11.26	11.81	2.16
Charging Completion	98.7%	98.3%	89.3%	91.1%	92.7%
Non-completion penalty ($)	3.52	5.06	67.92	32.35	19.88
Total cost ($)	26.88	29.95	83.36	45.89	42.61

To further demonstrate the effectiveness of the proposed approach, we compare the aggregation results of the proposed approach and model-driven method over 3 consecutive test days, as shown in Fig. 5.



(a) Flexibility interval of B1.

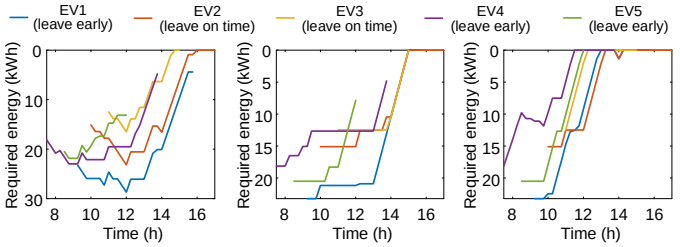
(b) Flexibility interval of B2.

(c) Flexibility interval of M-SAC.

Fig. 5. Aggregation results over 3 consecutive test days.

It can be noticed that the aggregation results of the three methods have different characteristics, and in general, the flexibility interval of B1 is the largest, that of M-SAC is the second one, and that of B2 is the most conservative. Method B1 obtains the flexibility interval by directly summing the energy and power boundaries of all EVs, which on one hand expands the original feasible region, and on the other hand, this method does not consider the impact on the future EV fleet when upward or downward flexibility is called up, and therefore leads to the charging demand of some EVs not being fulfilled during the test. Method B2 is the box-type inner approximation of the original feasible domain, which decouples the time periods and is able to cope with the uncertainty of regulation signals in different time periods, but this nature determines that the aggregator is biased towards providing downward flexibility, i.e., adjusting charging power upward, and the aggregation results is too conservative. M-SAC obtains moderately conservative aggregation results in a model-free manner without the need to predict EVs' departure times and regulation signals, and obtains as many flexibility reward as possible on top of ensuring that charging demand can be adequately met. It is also found that the base charging power is small (or even discharging) in the morning when the electricity price is high, and large in the noon when the price is low; since the reward rate of downward flexibility is larger than that of upward flexibility, the aggregator

reserves more upward flexibility than upward flexibility and concentrates it in the time period when the reward rate is high, which proves that the proposed method learns a good strategy.



(a) B1 method (b) B2 method (c) Proposed method

Fig. 6. EV charging process.

To verify the charging performance of the proposed method in an uncertain environment, 5 EVs are selected and their charging processes are shown in Fig. 6. It can be found that the proposed method has better performance in terms of charging completion compared to B1 and B2. B1 and B2 are basically able to fulfill the charging demand when the EVs leave according to the original schedule (like EV2 and EV3), but when the EV user picks up the vehicle early, B1 and B2 suffer from unfulfilled charging demand. This is due to the difficulty of modeling this uncertainty for B1 and B2 and when there is uncertainty in the DSO's regulation signals, it is even more challenging to predict the future energy state of EVs. In contrast, even with the presence of early departures, the charging demands of all 5 EVs are met in Fig. 6 (c). Therefore, the proposed method is adaptive to multiple uncertainties by learning effective strategies in a data-driven manner.

3) Advantage of the proposed real-time scheduling policy

To illustrate the advantage of the proposed scheduling policy, i.e., ILLLP, we compare it with existing real-time scheduling policies under the same M-SAC framework, including EDF and LLF [22]. The well-trained models are evaluated in the test set and the average daily performances are shown in Fig. 7.

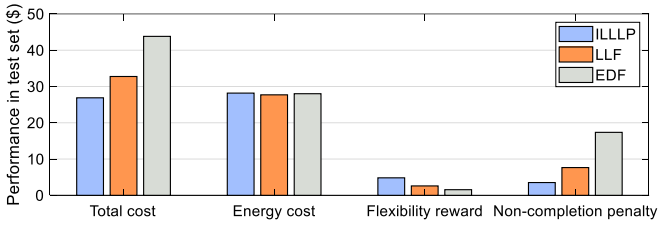


Fig. 7. Performance of different scheduling policies on the test set.

It can be found that although the three strategies are basically the same in terms of energy cost, there is a large disparity in other aspects, with the total cost of the proposed method decreasing by 17.9% and 38.7% compared to that of LLF and EDF, respectively. This is because EDF and LLF do not take into account the discharge mode, which results in a smaller power adjustable range of the EVCS and therefore lower flexibility rewards. In addition, the charging completion of ILLLP is also higher than the other two strategies, which proves the superiority of the proposed real-time scheduling policy, especially when there are uncertainties in the EV departure time and grid regulation signal, the proposed policy can satisfy more charging demands.

D. Scalability test

To verify the scalability of the proposed method, the simulation is also performed on the IEEE 118-bus radial distribution network with six EVCSs and each station has 500 charging bays [27]. Note that only one EVCS is selected as intelligent agent to implement the proposed method. Here, we compare two cases to verify the efficacy of introducing an auto-encoder:

Case 1: The proposed M-SAC algorithm, but auto-encoder is not included in the framework.

Case 2: The proposed M-SAC algorithm with auto-encoder for dimensionality reduction.

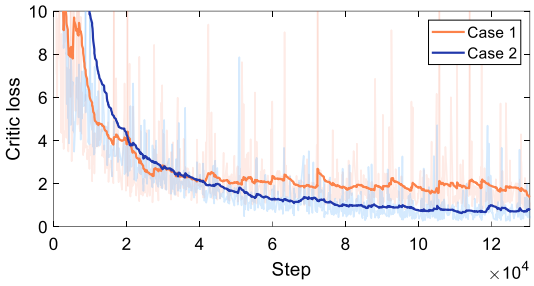


Fig. 8. Convergence process during the training.

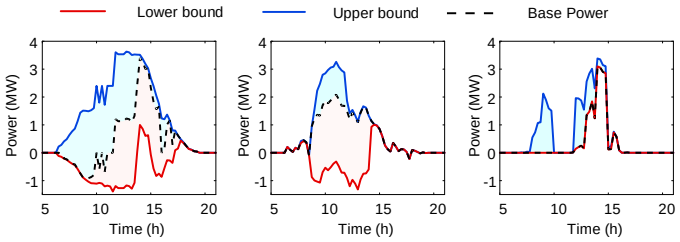
The training performance of two cases is shown in Fig. 8. It can be found that the critical loss in Case 2 reaches a lower level after about 6×10^4 update steps, which indicates that the agent is relatively accurate in estimating the Q-values of different actions and states, and the training time and moving average return at this slot are 4.11 hours and -5.79, respectively. In Case 1, then agent needs to reach the same level after 13×10^4 update steps, at which slot the training time and moving average return are 8.98 hours and -5.51, respectively. Therefore, on the basis of achiveving comparable training effect, the proposed method can effectively reduce the training time to better cope with larger-scale aggregation scenarios.

TABLE IV. Test performance comparison of different methods.

	M-SAC	B1	B2
Energy cost ($)	859.53	787.45	752.74
Flexibility reward ($)	196.68	385.73	85.57
Charging Completion	99.0%	93.3%	96.1%
Non-completion penalty ($)	129.74	649.82	377.27
Total cost ($)	792.60	1051.55	1044.44
Average time per decision (s)	0.001	0.22	24.35

Table IV compares the performance of the proposed method with the B1, B2 methods based on the same test setting. Fig. 9 shows the aggregation results of the three methods for a typical test day. Fig. 10 shows the distribution of all EVs that failed to complete 100% charging during the test days. It can be observed that although method B1 provides more flexibility to the grid and more revenue to the EVCS, it is unable to guarantee a high level of charging completion, with 8290 EVs, or about 19% of the total number of EVs, failing to reach 85% of the targeted charging volume; The aggregation results for method B2 are too conservative and still have a high number of EVs with low charging completions due to the lack of consideration of EVs departure uncertainty; The proposed approach can effectively deal with multiple uncertainties and balance economy and user

satisfaction, providing greater flexibility to the grid while meeting the charging demand of most EVs.



(a) B1 (b) M-SAC (c) B2

Fig. 9. Aggregated flexibility interval on a typical day.

In addition to the economic advantages, the average decision time of the proposed method is much lower than the other two methods. This is due to the fact that B1 and B2 are model-driven and need to solve the optimization model on a rolling process, and when the EV scale reaches a certain level, the decision time of B1 and B2 can hardly meet the real-time requirement. In contrast, the policy obtained by the proposed method is based on a neural network, and the decision time is only the time of forward pass, which is more suitable for dynamic aggregation of large-scale EVs in real time.

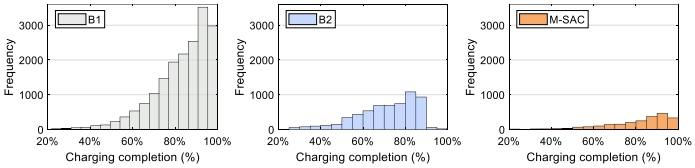


Fig. 10. Distribution of all EVs that failed to complete 100% charging.

VI. Conclusions

This paper addresses two significant issues in EVs cluster management: real-time dynamic aggregation and power allocation. The problem of optimal aggregate flexibility offering is formulated as an MDP and solved by the proposed M-SAC algorithm. To allocate the disaggregated power, a casual scheduling policy is proposed. The effectiveness of the proposed method is validated by simulations conducted on small and large distribution systems, and the main findings are listed as follows:

1) The proposed M-SAC algorithm shows superior training performance compared to the state-of-the-art DRL algorithms.

2) The proposed aggregation method is more adaptive to multiple uncertainties than the traditional model-driven methods, exhibiting better overall performance in terms of flexibility offering and user satisfaction.

3) The integration of dimensionality reduction process in the M-SAC algorithm can effectively accelerate the training speed in larger scale aggregation scenarios.

VII. References

[1] IEA (2023), World Energy Outlook 2023, IEA, Paris https://www.iea.org/reports/world-energy-outlook-2023.

[2] M. Zhou, Z. Wu, J. Wang and G. Li, "Forming dispatchable region of electric vehicle aggregation in microgrid bidding," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4755-4765, 2021.

[3] R. R. Appino, V. Hagenmeyer and T. Faulwasser, "Towards optimality preserving aggregation for scheduling distributed energy resources," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 3, pp. 1477-1488, 2021.

[4] X. Wang et al., "Tri-Level scheduling model considering residential demand flexibility of aggregated HVACs and EVs under distribution LMP," *IEEE Trans. Smart Grid*, vol. 12, no. 5, pp. 3990-4002, 2021.

[5] J. Hu, J. Wu, X. Ai and N. Liu, "Coordinated energy management of prosumers in a distribution system considering network congestion," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 468-478, 2021.

[6] Y. Wen, Z. Hu, S. You and X. Duan, "Aggregate feasible region of ders: exact formulation and approximate models," *IEEE Trans. Smart Grid*, vol. 13, no. 6, pp. 4405-4423, 2022.

[7] L. Zhao, W. Zhang, H. Hao and K. Kalsi, "A geometric approach to aggregate flexibility modeling of thermostatically controlled loads," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4721-4731, 2017.

[8] M. Wang et al., "State space model of aggregated electric vehicles for frequency regulation," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 981-994, 2020.

[9] S. Kiani, K. Sheshyekani and H. Dagdougui, "An extended state space model for aggregation of large-scale evs considering fast charging," IEEE Trans. Transp. Electrification, vol. 9, no. 1, pp. 1238-1251, 2023.

[10] X. Wu et al., "Heterogeneous aggregation and control modeling for electric vehicles with random charging behaviors," *IEEE Trans. Sustain. Energy*, vol. 14, no. 1, pp. 525-536, 2023.

[11] X. Chen, E. Dall'Anese, C. Zhao and N. Li, "Aggregate power flexibility in unbalanced distribution systems," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 258-269, 2020.

[12] D. Yan, S. Huang and Y. Chen, "Real-time feedback based online aggregate ev power flexibility characterization," *IEEE Trans. Sustain. Energy*, early access, 2023.

[13] Y. Zheng, Y. Song, D. J. Hill and K. Meng, "Online distributed mpc-based optimal scheduling for ev charging stations in distribution systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 638-649, 2019.

[14] H. Li, Z. Wan and H. He, "Constrained EV charging scheduling based on safe deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2427-2439, 2020.

[15] L. Yan et al., "Deep reinforcement learning for continuous electric vehicles charging control with dynamic user behaviors," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5124-5134, 2021.

[16] Z. Zhang et al., "Federated reinforcement learning for real-time electric vehicle charging and discharging control," *arXiv preprint* arXiv:2210.01452, 2022.

[17] S. Wang, S. Bi and Y. A. Zhang, "Reinforcement learning for real-time pricing and scheduling control in ev charging stations," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 849-859, 2021.

[18] Y. Cao, H. Wang, D. Li and G. Zhang, "Smart online charging algorithm for electric vehicles via customized actor-critic learning," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 684-694, 2022.

[19] N. Sadeghianpourhamami, J. Deleu and C. Develder, "Definition and evaluation of model-free coordination of electrical vehicle charging with reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 203-214, 2020.

[20] Y. Jiang et al., "Data-driven coordinated charging for electric vehicles with continuous charging rates: a deep policy gradient approach," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12395-12412, 2022.

[21] E. Wong, L. Kin, and F. Tony, "State-space decomposition for reinforcement learning." *Dept. Comput., Imperial College London*, London, UK, Rep, 2021.

[22] A. Subramanian et al., "Real-time scheduling of distributed resources," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2122-2130, 2013.

[23] Y. Xu, F. Pan and L. Tong, "Dynamic scheduling for charging electric vehicles: a priority rule," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4094-4099, 2016.

[24] N. Vieillard, O. Pietquin, M. Geist, "Munchausen reinforcement learning," Adv. Neural Inf. Process Syst., vol. 33, pp. 4235-4246, 2020.

[25] "Electric Chargepoint Analysis 2017: Domestics," UK Department for Transport, Tech. Rep., 2018. [Online]. Available: https://www.gov.uk/government/statistics/electric-chargepoint-analysis-2017-domestics

[26] W. Mai and C. Y. Chung, "Economic MPC of Aggregating Commercial Buildings for Providing Flexible Power Reserve," *IEEE Trans. Power Syst.*, vol. 30, no. 5, pp. 2685-2694, 2015.

[27] Zhang, Mingyang (2023). Simulation dataset for large-scale system. figshare. Dataset. https://doi.org/10.6084/m9.figshare.24750279.v1.