Constrained EV Charging Scheduling Based on Safe Deep Reinforcement Learning

Hepeng Li, *Student Member, IEEE*, Zhiqiang Wan, *Student Member, IEEE*, and Haibo He, *Fellow, IEEE*

***Abstract*—Electric vehicles (EVs) have been popularly adopted and deployed over the past few years because they are environment-friendly. When integrated into smart grids, EVs can operate as flexible loads or energy storage devices to participate in demand response (DR). By taking advantage of time-varying electricity prices in DR, the charging cost can be reduced by optimizing the charging/discharging schedules. However, since there exists randomness in the arrival and departure time of an EV and the electricity price, it is difficult to determine the optimal charging/discharging schedules to guarantee that the EV is fully charged upon departure. To address this issue, we formulate the EV charging/discharging scheduling problem as a constrained Markov Decision Process (CMDP). The aim is to find a constrained charging/discharging scheduling strategy to minimize the charging cost as well as guarantee the EV can be fully charged. To solve the CMDP, a model-free approach based on safe deep reinforcement learning (SDRL) is proposed. The proposed approach does not require any domain knowledge about the randomness. It directly learns to generate the constrained optimal charging/discharging schedules with a deep neural network (DNN). Unlike existing reinforcement learning (RL) or deep RL (DRL) paradigms, the proposed approach does not need to manually design a penalty term or tune a penalty coefficient. Numerical experiments with real-world electricity prices demonstrate the effectiveness of the proposed approach.**

***Index Terms*—Constrained Markov decision process, safe deep reinforcement learning, model-free, EV charging scheduling.**

I. Introduction

AS AN environment-friendly alternative to traditional fossil fuel-powered vehicles, EVs have been popularly adopted and deployed over the past few years [1]–[3]. According to the report of International Energy Agency (IEA), the number of EVs over the world reached about 3.1 million in 2017 [4]. It is expected that the number of EVs will grow to 125 million by 2030 [4]. Large-scale integration of EVs into the power grid can significantly stress the supply side, which will raise concerns about the potential impacts of frequency excursion, voltage fluctuation, and peak regulation. In order to alleviate these impacts, it is encouraged to shift the EV charging schedules to off-peak hours through DR [5]. In a DR program, EV charging schedules can be optimized in response to time-varying prices [6] to reduce the charging costs, or even make revenues by discharging energy to the grid [7].

However, it is challenging to efficiently manage EV charging schedules in real-time due to the existence of randomness. Specifically, influenced by traffic conditions and user's commuting behavior, the remaining energy, arrival time and departure time of an EV are unknown in advance. To guarantee the EV can be fully-charged before departure, a straightforward strategy is to finish the charging process as early as possible. However, in order to take advantage of the time-varying electricity prices, we hope the EV could be charged when the price is low and discharged when the price is high. The conflicts in these two objectives make it difficult to determine the optimal charging/discharging timing and energy quantity to satisfy the EV charging demand as well as minimize the charging cost.

Recently, RL has been widely used to make decisions under uncertain scenarios [8]–[10] because it can directly learn an optimal strategy from experience data, and there is no need to model the distribution of the randomness. The great success of RL inspires many researchers [11]–[13], [16] to develop RL based approaches for EV charging management. In [11], a Q-Table was implemented to approximate an action-value function that assessed the quality of the charging schedule. In order to implement Q-Table, the charging action and the electricity price was discretized. However, this discretization is not suitable for real-world application with a large number of actions and states since the Q-Table will become extremely large. In order to avoid the discretization step, Vandael *et al.* [12] proposed to approximate the action-value function with a set of linear basis functions. However, the linear approximator has limited capacity to handle the non-linear action-value function in real-world scenarios. Unlike this linear approximator, a non-linear kernel averaging regression operator was proposed in [13] to approximate the action-value function. The drawback is that the kernel function should be manually selected, and its parameters are also required to be properly designed. Inspired by the great success of deep neural network [14], [15], Wan *et al.* [16] used a deep neural network to approximate the action-value function and developed a real-time EV charging controller based on a DRL approach.

Although the aforementioned methods achieve promising results, they need to properly design a penalty term and choose the penalty coefficient to make sure the EV can be fully charged upon departure. The process of designing the coefficient is tedious. In addition, the performance of these

Manuscript received July 30, 2019; revised October 4, 2019 and November 19, 2019; accepted November 19, 2019. Date of publication November 22, 2019; date of current version April 21, 2020. This work was supported by the National Science Foundation under Grant ECCS 1917275. Paper no. TSG-01101-2019. *(Corresponding author: Haibo He.)*

The authors are with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, South Kingstown, RI 02881 USA (e-mail: hepengli@uri.edu; zwan@ele.uri.edu; he@ele.uri.edu).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2019.2955437

methods may deteriorate due to unsuitable penalty coefficient. To avoid the burden of manually choosing this coefficient, we formulate the real-time EV charging scheduling problem as a CMDP with a major concern about the constraint. With the CMDP formulation, we can guarantee to fully charge the EV upon departure and minimize the charging cost. To solve the CMDP, a completely model-free approach is proposed to find the constrained optimal EV charging/discharging scheduling strategy based on a SDRL paradigm, i.e., constrained policy optimization (CPO) [17]. Different from existing DRL methods, the proposed approach can directly handle the constraint and does not need to design a penalty term and choose a penalty coefficient for the constraint. The effectiveness of the proposed approach is validated through experimental studies.

Compared to our previous work [16], the main contribution of this paper is that we formulate the EV charging shedluling as a CMDP and propose a SDRL solution based on CPO to handle the charging constraint. The proposed approach can directly solve for the constrained optimal charging policy and does not need to design a penalty term for the charging constraint. Specifically, the contributions of this paper are threefold.

(i) A CMDP model is formulated for the constrained EV charging/discharging scheduling problem. The formulation considers the randomness of the EV's arrival time, departure time and remaining energy, as well as the real-time electricity price. The aim is to find the constrained charging/discharging scheduling strategy so that the charging cost can be minimized and the user's charging demand is satisfied.

(ii) A SDRL-based solution that does not require any knowledge about the randomness and the constraint is proposed to determine the constrained optimal charging and discharging schedules. Unlike existing RL or DRL paradigms, the proposed approach does not need to manually design the penalty term and tune the penalty coefficient for the constraint.

(iii) A DNN is designed to learn to generate constrained optimal charging/discharging schedules directly from raw state information of the EV and the electricity price in a completely end-to-end manner.

The rest of the paper is organized as follows. Section II presents the CMDP formulation of the EV charging/discharging scheduling problem. Section III proposed the SDRL-based approach to solve the CMDP. Then, in Section IV, numerical experiments are carried out to validate the effectiveness of the proposed approach. Finally, Section V draws the conclusions.

II. Problem Formulation

In this section, the constrained EV charging/discharging scheduling problem is formulated as a CMDP. The aim is to minimize the user's electricity cost as well as satisfy the EV charging demand. In the formulation, the randomness in the arrival time, departure time and remaining energy of the EV is considered. The uncertainty of the real-time electricity price is also taken into account. In the following subsections, we first define the problem as an MDP without considering the charging demand constraint. Then, the CMDP formulation for the problem is proposed by augmenting the MDP with the constraint.

A. MDP Formulation

The real-time EV charging/discharging scheduling problem can be defined as an MDP with a 5-tuple (S, A, P, R, γ), where S is the set of states; A is the set of actions; $P : S \times A \times S \to [0, 1]$ is the transition probability function; $R : S \times A \times S \to \mathbb{R}$ is the reward function; γ is a discounted factor, which balances the importance between the immediate reward and future rewards.

1) State: The state is defined as $s_t = (E_t, P_{t-23}, \ldots, P_t)$, $\forall t$, which contains two types of information: the EV battery energy E_t at time step t, and the past 24 hours' electricity prices $P_{t-23}, \ldots, P_t$. In this study, we assume that the EV user is a price-taker [18], [19] and the charging action does not affect the electricity price. The EV battery energy can be viewed as the physical state of our system, which is the physical resource we are managing. The electricity price can be viewed as the information state, which we need to make a decision and compute the objective function. Since the future electricity price is unknown, we use the past 24-hours electricity price to infer future price trends so that we can make the most cost-effective charging decisions.

2) Action: The action is the quantity of the charging or discharging energy at time step t. It is defined as a continuous variable $a_t \in [-e^{dis}_{max}, e^{ch}_{max}]$, where e^{ch}_{max} and e^{dis}_{max} represent the allowed maximum charging and discharging energy, respectively. When the EV is charged, the action a_t is positive. When the EV is discharged, the action a_t is negative.

3) Transition Probability: The transition probability $P(s'|s, a)$ is influenced by the charging action, dynamics of the EV battery and randomness of the electricity price. To formulate the real-world scenario, we consider the transition probability is unknown. For the purpose of simulation, we model the dynamics of the EV battery as $E_{t+1} = E_t + a_t - e_{loss}$, where e_{loss} is the energy loss during the charging and discharging process. The energy loss is model by $e_{loss} = a_t \cdot \eta_{ch}$ if $a_t \geq 0$ or $e_{loss} = a_t/\eta_{dis}$ otherwise, where η_{ch} and η_{dis} represent the energy conversion efficiency during charging and discharging, respectively.

4) Reward: From the users' perspective, the reward is formulated as $r_t = R(s_t, a_t, s_{t+1}) = -a_t * P_t, \forall t$. During the charging process, the reward denotes the negative of charging cost. During the discharging process, the reward represents the revenue from selling electricity to the grid. Here we assume that the selling price of the electricity is the same as the purchasing price as suggested by [16].

The aim is to select a policy π which maximizes the total discounted return,

$$J(\pi) = E_{\tau\sim\pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right], \tag{1}$$

where $\gamma \in [0, 1)$ is the discount factor, τ denotes a trajectory ($\tau = (s_0, a_0, s_1, \ldots)$), and π denotes the probability of selecting action a in state s.

B. CMDP Formulation

To consider the EV charging constraint, we define an auxiliary cost function $C : S \times A \times S \to \mathbb{R}$ by

$$c_t = \begin{cases} \left|E_t - E_{target}\right|, & if\ t = T, \\ E_t - E_{max}, & if\ E_t > E_{max},\ t < T, \\ E_{min} - E_t, & if\ E_t < E_{min},\ t < T \end{cases} \tag{2}$$

where the first line computes how much the EV battery energy deviates from its charging target E_{target} at the departure time T, the second line measures the amount of energy that exceeds its allowable maximum value E_{max}, and the third line calculates the amount of energy that is below its allowable minimum value E_{min}.

Let $J_C(\pi)$ denote the expected discounted return of the policy π with respect to the auxiliary cost C : $J_C(\pi) = E_{\tau\sim\pi}[\sum_{t=0}^{T} \gamma^t c_t]$ (referred to as C-return in the following content). Then, the MDP formulation can be augmented to handle the constraint by confining its policies to the following feasible set

$$\Pi_c = \{\pi : J_C(\pi) <= d\} \tag{3}$$

where d is a small tolerance for the charging constraint violation. Therefore, we have the following CMDP formulation for the EV charging/discharging scheduling problem,

$$\begin{aligned} \max_{\pi}\ & J(\pi) = E_{\tau\sim\pi}\left[\sum_{t=0}^{T} \gamma^t R_t\right] \\ s.t.\ & J_C(\pi) \le d. \end{aligned} \tag{4}$$

and the optimal policy π^* for the CMDP can be defined by

$$\pi^* = \arg\max_{\pi\in\Pi_C} J(\pi). \tag{5}$$

In conventional MDP framework, the constraint is generally formulated as a penalty term in the objective via a Lagrange multiplier as follows

$$\max_{\pi}\ J(\pi) - \varrho \cdot f(d, J_C(\pi)), \tag{6}$$

where $\varrho > 0$ is the Lagrange multiplier; $f(d, J_C(\pi))$ is the penalty function, which satisfies

$$\begin{aligned} & f(d, J_C(\pi)) = 0,\ \text{if } J_C(\pi) \le d, \\ & f(d, J_C(\pi)) > 0,\ \text{else if } J_C(\pi) > d. \end{aligned} \tag{7}$$

During the optimization, we want to minimize the penalty term $f(d, J_C(\pi))$ and maximize the return $J(\pi)$. To reach this aim, we need an appropriate value for Lagrange multiplier ϱ to keep a balance between the penalty $f(d, J_C(\pi))$ and the return $J(\pi)$. A small value for the Lagrange multiplier could cause inadequate penalization of the constraint violation. In this case, the EV would not be fully charged when it departed. On the contrary, a large value for the Lagrange multiplier could cause an excessive punishment over the constraint, resulting in less cost-effective charging schedules. In practice, tuning the Lagrange multiplier ϱ usually requires a tedious process of trial-and-error. Nevertheless, the proposed CMDP does not need to manually tune the Lagrange multiplier ϱ. In the next section, we propose a SDRL algorithm to solve the CMDP in a completely model-free manner.

III. Proposed Approach

In traditional RL or DRL paradigms, unconstrained MDPs are generally approached by policy search algorithms, which search for the optimal policy within a set Π_θ of parameterized policies π_θ. A typical policy search algorithm is policy gradient (PG), which searches for a local maximum in $J(\pi_\theta)$ by ascending the gradient of the logarithm of the policy π_θ with respect to the parameters θ,

$$\theta^{new} = \theta^{old} + \alpha\, \nabla_\theta J(\pi_\theta), \tag{8}$$

$$\begin{aligned} \nabla_\theta J(\pi_\theta) &= \sum_{s\in S} d(s) \sum_{a\in A} \pi_\theta(a|s)\, \nabla_\theta \log \pi_\theta(a|s) \Psi_t \\ &= E_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) \Psi_t\right], \end{aligned} \tag{9}$$

where $\nabla_\theta J(\pi_\theta)$ is the policy gradient, $d(s) = \lim_{t\to\infty} Pr\{s_t = s|s_0, \pi\}$ is the stationary distribution of states under π, and α is a step-size parameter; and Ψ_t can be discounted returns $\Psi_t = \sum_{k=1} \gamma^{k-1} R_{t+k}$ (REINFORCE algorithm) or action-values $Q^\pi(s, a)$ (Actor-Critic algorithm) or other formulas (resulting in different variants of the PG algorithm).

However, for the formulated CMDP, the policy π_θ is constrained. Therefore, instead of searching in Π_θ, we need to optimize over $\Pi_C \cap \Pi_\theta$:

$$\begin{aligned} \pi_{\theta^{new}} = \arg\max_{\pi_\theta}\ & J(\pi_\theta) \\ s.t.\ & J_C(\pi_\theta) \le d. \end{aligned} \tag{10}$$

This update is difficult to implement for conventional DRL approaches because it requires evaluation of the constraint function to determine whether a proposed policy $\pi_{\theta^{new}}$ is feasible.

A. Constrained Policy Update Rule

To solve the CMDP, we introduce Theorem 1 and Corollary 1 and 2 from [17] in the following contents, which provide guidance for the safe update of the policy to maximize the expected return and satisfy the charging constraint.

Theorem 1: For any function $f : S \to \mathbb{R}$ and any policies π' and π, define $\delta_f(s, a, s') = R(s, a, s') + \gamma f(s') - f(s)$, $\epsilon_f^{\pi'} = \max_s |E_{a\sim\pi', s'\sim P}[\delta_f(s, a, s')]|$

$$L_{\pi,f}(\pi') = E_{s\sim d^\pi, a\sim\pi, s'\sim P}\left[\left(\frac{\pi'(a|s)}{\pi(a|s)} - 1\right)\delta_f(s, a, s')\right], \tag{11}$$

$$D^{\pm}_{\pi,f}(\pi') = \frac{L_{\pi,f}(\pi')}{1-\gamma} \pm \frac{\sqrt{2}\gamma\epsilon_f^{\pi'}}{(1-\gamma)^2}\sqrt{E_{s\sim d^\pi}[D_{KL}(\pi'||\pi)[s]]}, \tag{12}$$

where $D_{KL}(\pi'||\pi)[s]$ is the total KL-divergence between the policies π' and π at state s

$$D_{KL}(\pi'||\pi)[s] = -\int_{a\in A} \pi'(a|s) \log\left(\frac{\pi'(a|s)}{\pi(a|s)}\right) da. \tag{13}$$

Then, the following bounds hold:

$$D^{+}_{\pi,f}(\pi') \ge J(\pi') - J(\pi) \ge D^{-}_{\pi,f}(\pi'). \tag{14}$$

Corollary 1: For any two policies π', π, given $\epsilon^{\pi'} = \max_s |E_{a\sim\pi'} A^{\pi}(s,a)|$, the following bound holds:

$$J(\pi') - J(\pi) \geq \frac{1}{1-\gamma} E_{\substack{s\sim d^{\pi} \\ a\sim\pi'}} \left[A^{\pi}(s,a) - \frac{\sqrt{2}\gamma\epsilon^{\pi'}}{1-\gamma} \sqrt{D_{KL}(\pi'||\pi)[s]} \right] \quad (15)$$

where $A^{\pi}(s,a)$ represents advantage function, which is calculated by $A^{\pi}(s,a) = R(s,a,s') + \gamma V^{\pi}(s') - V^{\pi}(s)$, and $V^{\pi}(s)$ is the value function.

Corollary 2: For any two policies π', π, and any cost function C, given $\epsilon_C^{\pi'} = \max_s |E_{a\sim\pi'} A_C^{\pi}(s,a)|$, the following bound holds:

$$J_C(\pi') - J_C(\pi) \leq \frac{1}{1-\gamma} E_{\substack{s\sim d^{\pi} \\ a\sim\pi'}} \left[A_C^{\pi}(s,a) + \frac{\sqrt{2}\gamma\epsilon_C^{\pi'}}{1-\gamma} \sqrt{D_{KL}(\pi'||\pi)[s]} \right] \quad (16)$$

where $A_C^{\pi}(s,a)$ denotes the advantage function with respect to the constraint. It is calculated by $A_C^{\pi}(s,a) = R(s,a,s') + \gamma V_C^{\pi}(s') - V_C^{\pi}(s)$, and $V_C^{\pi}(s)$ is the constraint value function.

We refer the readers to the reference in [17] for the proof of Theorem 1. The proof of Corollary 1 and 2 is given in Appendix A. To connect the theoretical results to our problem, let us substitute π and π' in Corollary 1 and 2 with $\pi_{\theta k}$ and $\pi_{\theta k+1}$, respectively, where $\pi_{\theta k}$ denotes the charging policy at the kth iteration and $\pi_{\theta k+1}$ denotes the charging policy at the $k+1$th iteration. Then, based on the results in (15) and (16), it follows that the following update rule (explained in Appendix B)

$$\begin{aligned} \pi_{k+1} = \arg\max_{\pi\in\Pi_\theta} \; & E_{\substack{s\sim d^{\pi_k} \\ a\sim\pi}} \left[A^{\pi_k}(s,a) \right] - \alpha_k \sqrt{\overline{D}_{KL}(\pi||\pi^k)[s]} \\ s.t. \; & J_C(\pi_k) + E_{\substack{s\sim d^{\pi_k} \\ a\sim\pi}} \left[\frac{A_C^{\pi_k}(s,a)}{1-\gamma} \right] \\ & + \beta_k \sqrt{\overline{D}_{KL}(\pi||\pi_k)[s]} \leq d \end{aligned} \quad (17)$$

is guaranteed to generate a monotonically nondecreasing sequence of policies that satisfy the safety constraint. Here α_k and β_k are proper coefficients to penalize the KL-Divergence $\overline{D}_{KL}(\pi'||\pi_k) = E_{s\sim d^{\pi^k}}[D_{KL}(\pi||\pi^k)[s]$ of the current policy π_k and the updated one π'. However, penalizing the policy divergence between π_k and its update π in the objective and the constraint could result in small step sizes and slow convergence. Instead, we can restrict the KL divergence $\overline{D}_{KL}(\pi'||\pi_k)$ by a trust region d to enable larger step sizes, as suggested by [20]. Therefore, we derive the following safe policy update rule

$$\begin{aligned} \pi_{k+1} = \arg\max_{\pi\in\Pi_\theta} \; & E_{\substack{s\sim d^{\pi_k} \\ a\sim\pi}} \left[A^{\pi_k}(s,a) \right] \\ s.t. \; & J_C(\pi_k) + \frac{1}{1-\gamma} E_{\substack{s\sim d^{\pi_k} \\ a\sim\pi}} \left[A_C^{\pi_k}(s,a) \right] \leq d, \\ & \overline{D}_{KL}(\pi||\pi_k) \leq \delta. \end{aligned} \quad (18)$$

B. DNN-Based Policy

In the proposed approach, we optimize the policy π with a Gaussian distribution $\pi_\theta \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$, where the expectation μ_θ and logarithmic standard deviation $\log\sigma_\theta$ of the Gaussian distribution $\mathcal{N}(\mu_\theta, \sigma_\theta^2)$ is approximated by an multi-layer perceptron (MLP). The parameter θ is the set of the network weights of the MLP. The MLP model is expressed by

$$\begin{aligned} \mu_\theta &= W_\mu^T \cdot f + b_\mu, \\ \log\sigma_\theta &= W_\sigma^T, \end{aligned} \quad (19)$$

where $W_\mu, W_\sigma, b_\mu \in \theta$ are the output layer's weights and bias of the MLP, respectively, and f are the latent features extracted by the hidden layers of the MLP. The latent features f are calculated by

$$\begin{aligned} f &= ReLU\left(W_n^T \cdot v_n + b_n\right), \\ v_{l+1} &= ReLU\left(W_l^T \cdot v_l + b_l\right), \; l = 1, 2, \ldots, n-1, \\ v_1 &= s, \end{aligned} \quad (20)$$

where $W_l, b_l \in \theta, l = 1, 2, \ldots, n$ are the weighs and biases in the lth hidden layers, respectively; $ReLU(\cdot)$ is the Rectified Linear Units activation function, and s is the input of the MLP, i.e., the system state. We refer to the MLP as policy network in the following content.

During the training process, the π_θ generates action by drawing a sample from the Gaussian distribution $\mathcal{N}(\mu_\theta, \sigma_\theta^2)$ to explore the action space. In test cases, the policy π_θ takes the expectation value μ_θ, which is approximated by the well-trained MLP, as its action. It is noted that the proposed policy π_θ can handle continuous charging actions.

Since we need to calculate the advantages $A^{\pi}(s,a)$ and $A_C^{\pi}(s,a)$ for the policy update, we use another MLP that shares the same architecture as (20) to extract latent features f' and approximate the value functions $V^{\pi}(s)$ and $V_C^{\pi}(s)$ by a linear combination of f',

$$\left[V^{\pi}(s|\theta_v), V_C^{\pi}(s|\theta_v)\right]^T = W_V^T \cdot f' + b_V, \quad (21)$$

where θ_v are the parameters of the value function approximator, which is referred to as value network. The value network is optimized by

$$\theta_v^{k+1} = \theta_v^k + \beta \bigtriangledown_{\theta^k} E_{\substack{s\sim d^{\pi} \\ a\sim\pi}} \left[\left(V^{\pi}\left(s|\theta_v^k\right) - \sum_{l=0}^{\infty} \gamma^l R(s,a) \right)^2 + \left(V_C^{\pi}(s|\theta_v) - \sum_{l=0}^{\infty} \gamma^l C(s,a) \right)^2 \right], \quad (22)$$

where β is a step size parameter.

C. Constrained Policy Optimization Algorithm

The policy network could have tens of thousands of parameters, so directly optimizing the parameterized policy π_θ by the update rule (18) can be impractical due to serious nonlinearity and computational cost of the neural network. Nevertheless, the update (18) is well-approximated around θ_k by linearization of the objective and the safety constraint and by second order expansion of the KL divergence. Specifically, by taking Taylor series expansion, the update rule (18) can be

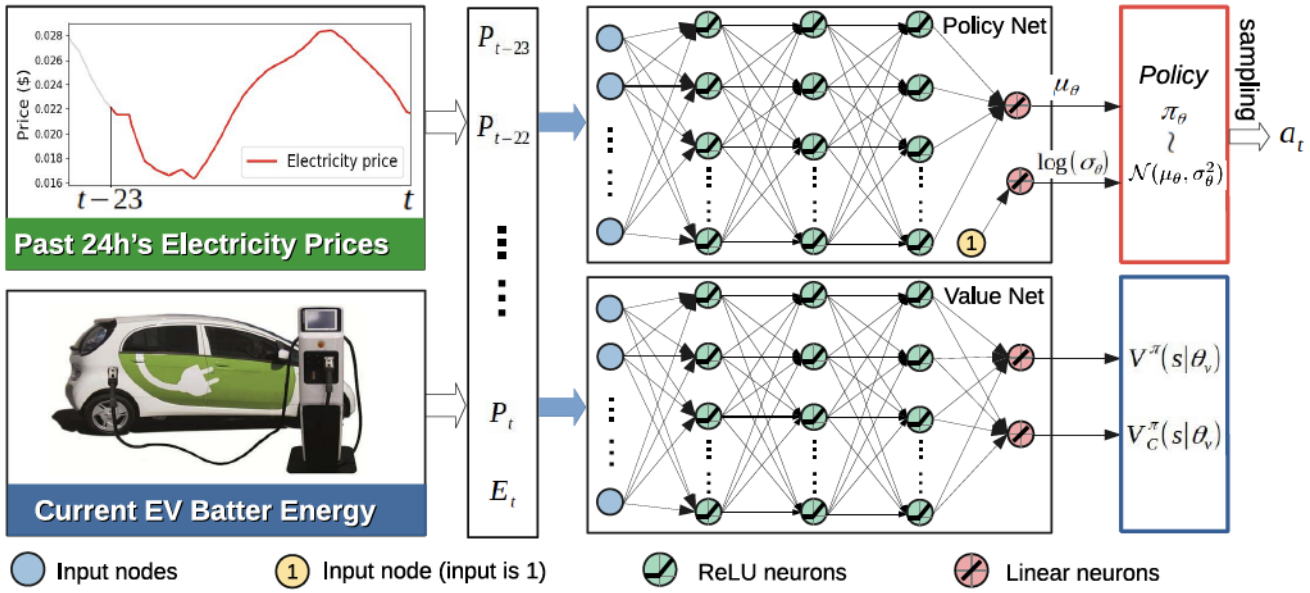


Fig. 1. Overall architecture of the designed policy network and the value network. The inputs of the both networks are the system state, i.e., the past 24h's electricity prices $P_{t-23}, \ldots, P_t$ and the current EV battery energy E_t. The policy network extracts features from the state information and outputs the mean values μ_θ and logarithmic standard deviations $\log(\sigma_\theta)$ of a normal distribution. By sampling from the normal distribution, the policy π_θ generates a charging/discharging action a_t for the EV. The value network shares the same architecture with the policy network. It extracts features from the state information and outputs the state-values $V^\pi(s|\theta_v)$ and $V_C^\pi(s|\theta_v)$.

approximated by (see Appendix C),

$$\begin{aligned}\theta^{k+1} = \arg\max_{\theta} \ & g^T(\theta - \theta_k) \\ s.t. \ & c + b^T(\theta - \theta_k) \le 0, \\ & \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k) \le \delta, \end{aligned} \tag{23}$$

where $g = \nabla_\theta E_{\substack{s\sim d^{\pi_{\theta^k}} \\ a\sim\pi_\theta}}[A^{\pi_{\theta^k}}(s,a)]$ is the first derivative of the objective in (18), $b = \nabla_\theta E_{\substack{s\sim d^{\pi_{\theta^k}} \\ a\sim\pi_\theta}}[\frac{A_C^{\pi_{\theta^k}}(s,a)}{1-\gamma}]$, $H = \nabla_\theta^2 \overline{D}_{KL}(\pi_\theta || \pi_{\theta^k})$, and $c = J_C(\pi_{\theta^k}) - d$.

The primary motivation for the update (23) is that it is easier to solve in practice than (18). Since the hessian matrix H of the KL-divergence is always positive semi-definite, the update rule (23) is a convex quadratic optimization and can be solved analytically with a guarantee of global optimum. By solving the optimization problem (23), we obtain the following safe policy update

$$\theta^{k+1} = \theta^k + \theta^* = \theta^k + \frac{1}{\lambda^*} H^{-1}(g - bv^*) \tag{24}$$

where θ^* is the optimal solution of the primal problem (23), and λ^*, v^* are the optimal solution of the dual problem of (23). Due to approximation error, the update rule (24) may result in an update that does not satisfy the true constraint $J_c(\pi_\theta^{k+1}) \le d$. For this issue, we use the following update in practice

$$\theta^{k+1} = \theta^k + \alpha \frac{1}{\lambda^*} H^{-1}(g - bv^*) \tag{25}$$

where the step size α is determined by a backtracking line search method to ensure the satisfaction of the constraint. In addition, the update rule (25) may sometimes be infeasible due to sampling error or bad update. For that case, we use the following update rule as suggested by [17]

$$\theta^{k+1} = \theta^k - \alpha \sqrt{\frac{2\delta}{b^T H^{-1} b}} H^{-1} b \tag{26}$$

to purely decrease the C-return $J_C(\pi^k)$ value.

To implement the update rule given in (23), we need to know the value of g, b, H and c. In practice, we can estimate these values at the kth iteration by (see Appendix D),

$$\widehat{g} = \frac{1}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{\nabla_\theta \pi_\theta(a_{n,m}|s_n)}{\pi_{\theta^k}(a_{n,m}|s_n)} A^{\pi_{\theta^k}}(s_n, a_{n,m}) \tag{27a}$$

$$\widehat{b} = \frac{1}{NM(1-\gamma)} \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{\nabla_\theta \pi_\theta(a_{n,m}|s_n)}{\pi_{\theta^k}(a_{n,m}|s_n)} A_C^{\pi_{\theta^k}}(s_n, a_{n,m}) \tag{27b}$$

$$\widehat{H} = \frac{1}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{\nabla_\theta \pi_\theta(a_{n,m}|s_n) \nabla_\theta^T \pi_\theta(a_{n,m}|s_n)}{\pi_{\theta^k}(a_{n,m}|s_n)} \tag{27c}$$

$$\widehat{c} = \frac{1}{D} \sum_{d=1}^{D} \sum_{t=0}^{T} \gamma^t c_t - d, \tau \sim \pi_{\theta^k} \tag{27d}$$

where $s_n \sim d^{\pi_{\theta^k}}, n = 1, \ldots, N$ are N sampled states at the kth iteration; $a_{n,m}, m = 1, \ldots, M$ are the M sampled actions from state s_n following the policy π_{θ^k}; D is the total number of trajectories $\tau_d = (s_0, a_0, s_1, \ldots)_d, d = 1, \ldots, D$ that are sampled at the kth iteration.

The CPO algorithm is summarized in Algorithm 1. At the beginning, the algorithm initializes the policy network parameters to θ^0, the value network parameters to θ_v^0, the maximum iterations to K, the trust region of the KL-divergence to δ and the trajectory buffer $\mathcal{D}$ size to D. Then, the algorithm goes into its main loop. In each loop, the algorithm samples a set of trajectories $\mathcal{D} = \{\tau_d\}_{d=1}^D$ following the policy

Algorithm 1 Safe Policy Update by CPO

1: **Inputs:** Initialized $\theta^0, \theta_v^0, K, \delta, D$.
2: **for** k = 1, K **do**
3: **for** d = 1, D **do**
4: Set time step counter $t \rightarrow 0$;
5: Reset the arrival time and the state s_t of the EV;
6: **while** t **is not** the departure time **do**
7: Sample an action a_t according to $\pi_\theta^k(a|s_t)$;
8: Observe the next state s_{t+1};
9: Calculate reward r_t and auxiliary cost c_t;
10: Set $t \rightarrow t+1$;
11: **end while**
12: Store trajectory $\tau_d = (s_0, a_0, r_0, c_0, s_1, \dots)$ in $\mathcal{D}$;
13: **end for**
14: calculate the sample estimates $\widehat{g}, \widehat{b}, \widehat{H}, \widehat{c}$;
15: **if** the optimization problem (23) is feasible **then**
16: Sovle (23) by $\theta^* = \frac{1}{\lambda^*} H^{-1}(g - bv^*)$;
17: Update θ^k by (25) via backtracking line search;
18: **else**
19: Update θ^k by (26) via backtracking line search;
20: **end if**
21: Update value network parameters θ_v^k by (22);
22: **end for**
23: **Output:** Optimal parameterized policy π_{θ^K}.

π_{θ^k} as shown in lines 3–13. After that, the sample estimates $\widehat{g}, \widehat{b}, \widehat{H}, \widehat{c}$ are calculated by using the sampled trajectories in $\mathcal{D}$. Starting from line 15, the algorithm checks whether the optimization problem (23) is feasible. If it is feasible, solving (23) by $\theta^* = \frac{1}{\lambda^*} H^{-1}(g - bv^*)$, and updating the policy by (25) via backtracking line search. Otherwise, update the policy by (26). Then, the value network parameters θ_v^k is updated by (22). When the loop ends, the algorithm outputs the optimal parameterized policy π_{θ^K}.

IV. Experimental Results

A. Experimental Setup

To validate the proposed approach, we use real-world electricity price data from [21]. The electricity price data are hourly time-varying retail prices that reflect the hourly wholesale market price for the Midcontinent Independent System Operator (MISO) delivery point. We use one-year data of 2017 as the training dataset and one-year data of 2018 as the test dataset. In addition, we assume that the EV user's driving behavior follows a certain pattern, which is widely used by researchers [22]–[25]. The assumption is reasonable because regular EV users have predictable habits and relatively fixed arrival and departure time, such as going to work in the morning and going back home in the evening. In our study, we model EV's arrival and departure time as truncated normal distributions as suggested by [25]. Table I presents the parameters of the distributions in details. For the arrival time, the mean and standard deviation are 18 and 1, respectively. It is bounded by [15, 21]. For the departure time, the mean and standard deviation are 8 and 1, respectively; and it is bounded by [6, 11]. For the remaining battery energy when the EV arrives home, we assume its mean and standard deviation are 50% and 10% of the capacity of the battery, respectively. In our experiments, we consider a Nissan Leaf EV with a maximum

TABLE I
Distributions Related to User's Commuting Behavior

	Distribution	Boundaries
Arrival Time	$\mathcal{N}(18, 1^2)$	$[15, 21]$
Departure Time	$\mathcal{N}(8, 1^2)$	$[6, 11]$
Remaining Energy[1]	$\mathcal{N}(0.5C, (0.1C)^2)$	$[0.2C, 0.8C]$

[1] C: Capacity=24kWh.

TABLE II
HyperParameters Used in Our Experiments

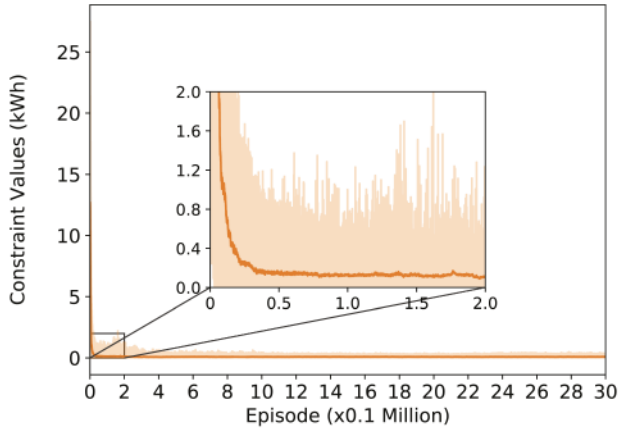
Notion	Value	Description
K	6000	maximum iterations
D	500	trajectory buffer size
γ	0.995	reward discount factor
d	0.1	constraint tolerance
δ	0.01	trust region of the KL-divergence
α	0.8^i	line search stepsize, $i = 0, 1, 2, \dots$
β	0.001	learning stepsize of value network

battery capacity of *Capacity* = 24 kWh. The allowable minimum and maximum energy of the battery are $E_{min} = 2.4$ kWh and $E_{max} = 24$ kWh, respectively. The maximum allowable charging and discharging energy at each hour are both 6 kwh. This means the action a_t can be chosen in the range $[-6, 6]$, and a negative value denotes discharging while a positive value represents charging. The coefficients of the energy conversion efficiency during charging and discharging are set to $\eta_{ch} = \eta_{dis} = 0.98$.

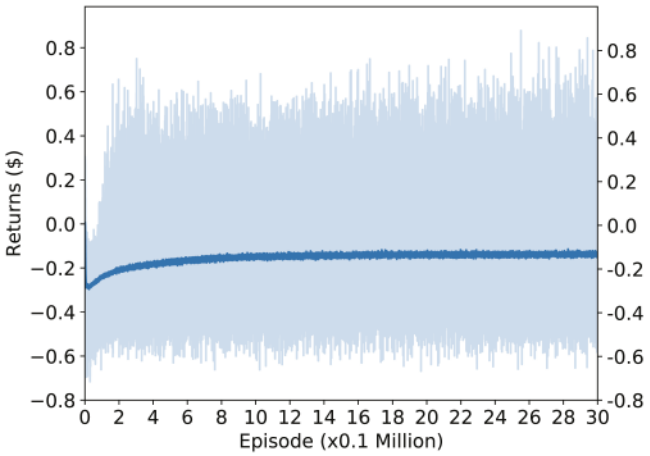
The charging energy target E_{target} is set to 24 kWh in order to fully charge the EV upon departure. The constraint tolerance d is set to 0.1 kWh. To learn a constrained optimal policy, the DNN-based policy π_θ modeled in Section III has 3 layers and each layer consists of 64 neuron units. The value network has the same architecture as the policy network. All network parameters are orthogonally initialized. The policy network parameters are updated by the proposed CPO algorithm and the value network parameters are updated by Adam gradient descent during the training process. We update the policy and the value network for K = 6000 iterations. At each iteration, we sampled D = 500 trajectories (or episodes) to prepare data to update the networks. Other parameter settings used in our experiments are presented in Table II. The experiments are conducted on a workstation with an NVIDIA TITAN Xp GPU and one i7-6800K CPU. The code is written in Python3.6 using the deep learning package TensorFlow1.12.

B. Baseline Methods

Before going into the evaluation of the proposed approach, we define several baseline methods for the purpose of comparison. First, we consider two well-known DRL methods, i.e., Deep Q-Network (DQN) and Deep Deterministic Policy Gradient (DDPG). Both methods can solve complex MDPs with high-dimensional state inputs, but they cannot be directly used to solve CMDPs. To make these methods able to handle the constraint, we add a penalty term in the objective.

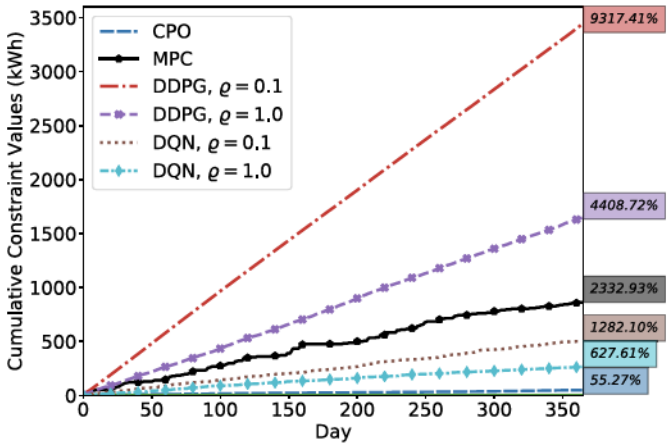


(a) Episode constraint values (light orange line) and the moving average (dark orange line).

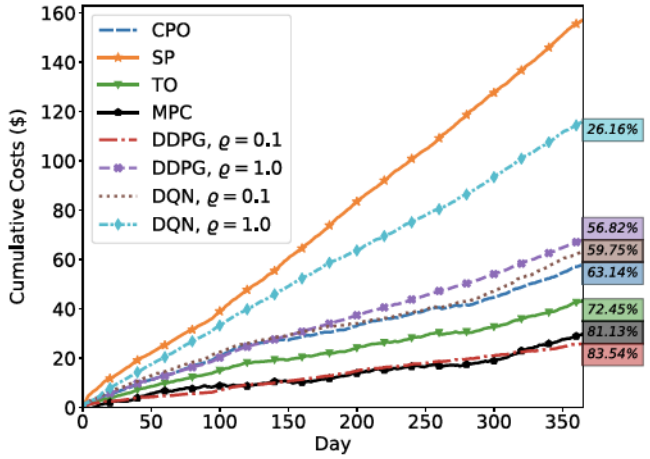


(b) Episode returns (light blue line) and the moving average (dark blue line).

Fig. 2. Constraint values and returns of the proposed approach during the training process.



(a) Cumulative curves of the daily constraint values.



(b) Cumulative curves of the daily electricity costs.

Fig. 3. Comparison of the cumulative curves of the daily electricity costs and the daily constraint values on the testset between the proposed approach and the baselines.

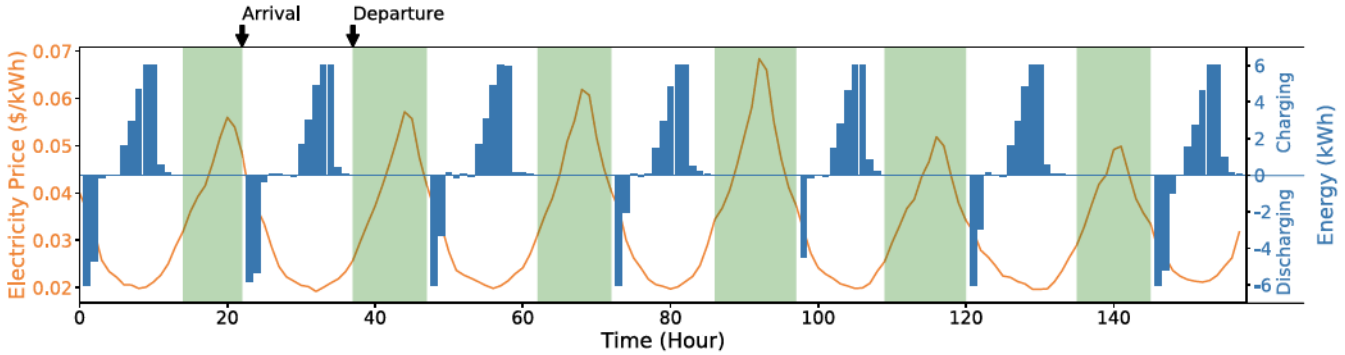
Specifically, we re-model the reward r_t as

$$r_t^c = r_t + \varrho \cdot c_t = -a_t P_t - \varrho \cdot \begin{cases} \left|E_t - E_{target}\right|, & if\ t = T, \\ E_t - E_{max}, & if\ E_t > E_{max},\ t < T, \\ E_{min} - E_t, & if\ E_t < E_{min},\ t < T \end{cases} \quad (28)$$

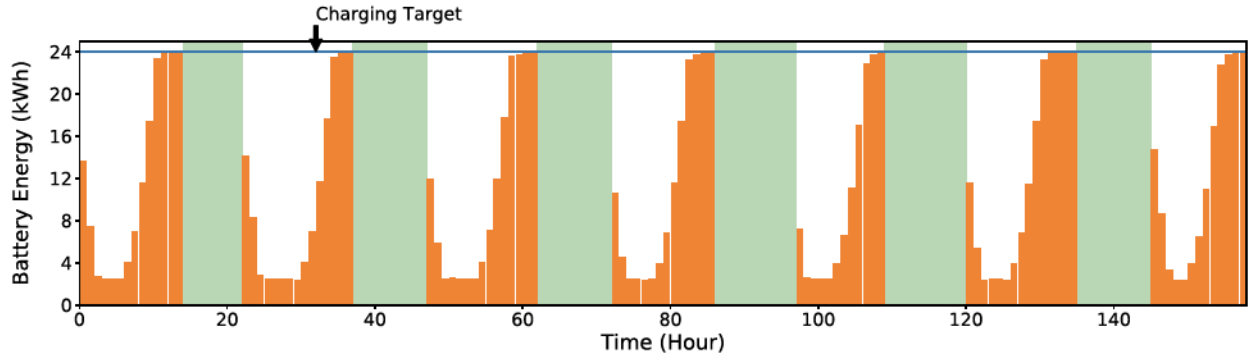
where ϱ is the penalty coefficient, or the so-called Lagrange multiplier. It is worth mentioning that the main difficulty of using DQN or DDPG is the determination of the penalty coefficient ϱ. In our study, we use two different values for the penalty coefficient, i.e., $\varrho = 0.1$ and 1.0, to conduct experiments, respectively. Then, we compare their performances with that of the proposed approach. In order for a fair comparison, we use the same architectures of the actor network and critic network for DDPG as those of the policy network $\pi_\theta(a|s)$ and value network $V^\pi(s|\theta_v)$ for the proposed approach. Since DQN cannot handle continuous actions, we discretize the action of the formulated model into 7 separate values (−6kW, −4kW, −2kW, 0kW, 2kW, 4kW, 6kW). We use the same architecture of the Q-network for DQN as that of the policy network $\pi_\theta(a|s)$ except that the dimensionality of the Q-network's output is 7.

Additionally, we deign another two baseline methods. The first baseline applies a "Safety-Prissy" (SP) strategy that charges the EV immediately with the maximum charging rate as soon as it arrives home and never discharges it. The second baseline assumes that all the uncertainties, including the arrival time, the departure time, the remaining energy of the EV, and the real-time electricity prices, are all known in advance. In this baseline, the EV charging/discharging scheduling problem is modeled as a deterministic optimization problem, and solved by SCIP [26]. We refer to this baseline as "Theoretical-Optimum" (TO). The TO baseline provides an upper limit for the performance but it cannot be reached in practice due to the existence of randomness. Both of the baselines can guarantee the satisfaction of the constraint.

Finally, we consider an optimization-based baseline method using model predictive control (MPC). At each time step, the MPC method forecasts the EV departure time and the future electricity price. Based on the forecasts, an optimization model is solved to derive the EV charging schedules, and only the



(a) Hourly electricity price (orange line) and the EV charging/discharging schedules (blue bar).



(b) Battery energy of the EV

Fig. 4. Scheduling results obtained by the proposed approach over 7 consecutive days (Jul 7th, 2018 - Jul 16th, 2018) in the test dataset. The areas covered by green in the subfigures represent the periods of time when the EV is absent from home.

schedule at the first step is executed. Then, the charging proceeds to the next step. The procedure is repeated until the EV departs. We assume that the distribution of the departure time is known by the MPC. The MPC predicts the departure time by drawing a sample from the distribution. For the electricity price, we assume that the forecasting error is 10 percent of the real electricity price. To generate the forecast data of the electricity price, we use the real electricity prices plus forecasting errors. The forecasting errors are sampled from the distributions $\mathcal{N}(0, 0.1P_t)$, $t = 1, 2, \ldots$, where P_t is the real electricity price at time step t.

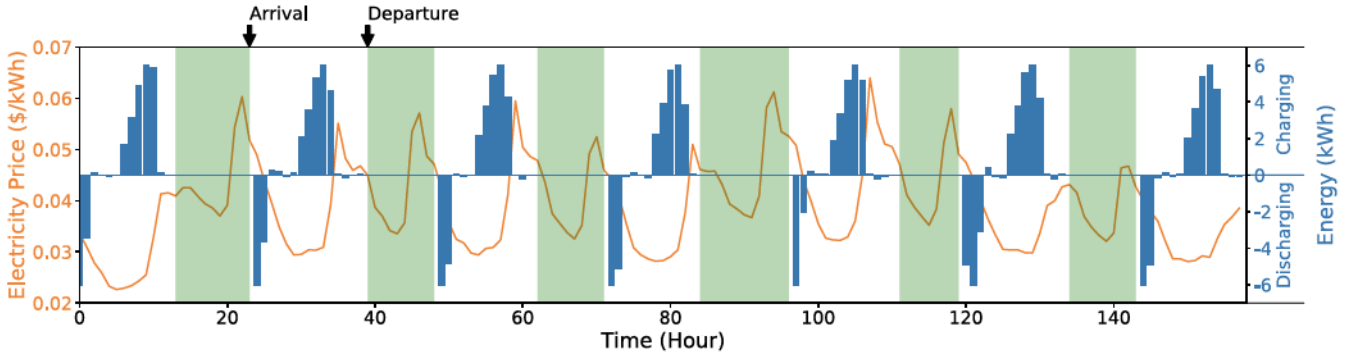
C. Numerical Results

Fig. 2 presents the constraint values and returns of the proposed approach during the training process. The constraint value $\sum_{t=0}^{T} \gamma^t c_t$ and return $\sum_{t=0}^{T} \gamma^t r_t$ in each episode are depicted by light orange curve in Figure 2a and light blue curve in Figure 2b, respectively. The corresponding moving average of the episode constraint value and return are depicted by dark orange curve in Figure 2a and dark blue curve in Figure 2b, respectively. As shown in Figure 2a, the episode constraint value decreases to a small region around the constraint tolerance shortly after the start of training. In addition, as shown in Figure 2b, the episode return gradually increases during the training process. When the training finishes, the moving average of the returns stabilizes at a point of convergence around -0.14. These results demonstrate that the proposed CPO approach is successful at learning to maximize the return and approximately enforcing the constraint for the formulated CMDP.

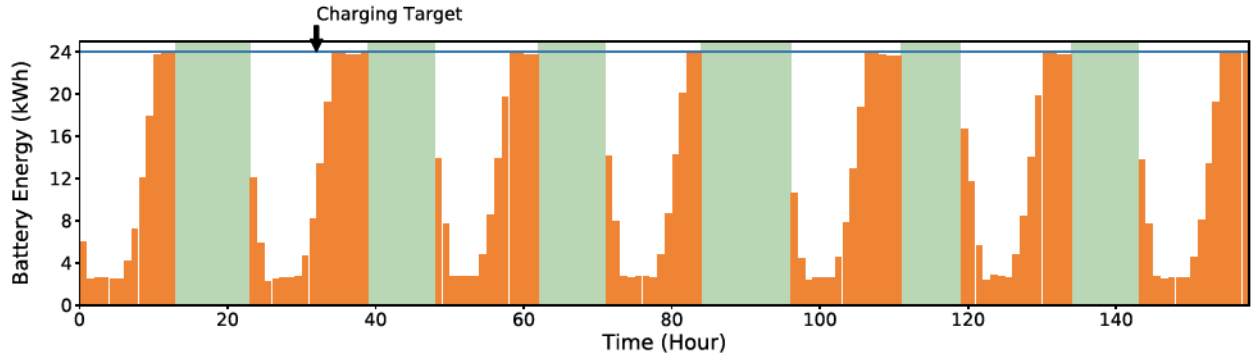
The proposed approach is then evaluated on the test dataset and compared with the baseline methods. Fig. 3 presents the comparison results. In the comparison, we calculate the daily constraint value $\sum_{t=0}^{T} c_t$ and electricity cost $\sum_{t=0}^{T} a_t P_t$ over the test days. Then, their cumulative curves are compared. Fig. 3a presents the cumulative curves of the daily constraint values of the proposed approach and the baseline methods. In this figure, small constraint values represent better performance. Since the TO and SP baselines can completely satisfy the constraint, their constraint values are 0 kWh. The percentage terms on the right illustrate the constraint violation ratio of the corresponding solution with respect to the constraint tolerance, which is calculated by

$$\frac{1}{N} \sum_{N} \left[\max\left(0, \sum_{t=0}^{T} c_t - d\right) / d \right] \times 100\%,$$

where $\sum_{t=0}^{T} c_t$ is the constraint value of the corresponding solution, $d = 0.1$ kWh is the constraint tolerance; $N = 365$ is the total number of the test days. As it can be seen in the figure, for the proposed CPO approach, the constraint violation ratio is only 55.27%, which means that on average the unsatisfied charging energy is only $55.27\% \times d = 0.05527$ kWh. However, for the DDPG and DQN baselines with the penalty coefficient ϱ equal to 0.1 and 1.0, the constraint violation ratios can



(a) Hourly electricity price (orange line) and the EV charging/discharging schedules (blue bar).



(b) Battery energy of the EV

Fig. 5. Scheduling results obtained by the proposed approach over 7 consecutive days (Dec 2nd, 2018 - Dec 9th, 2018) in the test dataset. The areas covered by green in the subfigures represent the periods of time when the EV is absent from home.

be high up to 9317.41%, 4408.72%, 1282.10% and 627.61% respectively. For the MPC baseline method, the constraint violation ratio is 2332.93%, which is also much higher than the CPO approach. It should be noted that because the departure time is random, it is impossible to completely satisfy the constraint. However, the proposed approach can effectively restrict the constraint violation so that the user's charging demand is adequately satisfied.

Fig. 3b presents the cumulative curves of the daily charging costs resulted from the proposed approach and the baseline methods. The percentage terms on the right represent the ratio of the cost reduction obtained by the corresponding solutions compared to the charging cost of the SP baseline method. As shown in the figure, the proposed CPO approach reduces the total charging cost by 63.14% in comparison with the SP baseline. The cost reduction is only 9.31% less than that of the TO baseline. It is worth noting that the result of the TO is 72.45% and cannot be reached. In addition, the proposed CPO approach is better than the baselines, i.e., the DDPG with $\varrho = 1.0$ and the DQN with $\varrho = 0.1$ and 1.0. These three baselines only reduce the charging cost by 56.82%, 59.75% and 26.16%, respectively. It should be noted that although the baseline of the DDPG with $\varrho = 0.1$ reduces the electricity cost by 83.54%, it significantly violates the constraint by a ratio of 9417.41% as shown in Fig. 3a. Considering the comparison results in Fig. 3a and Fig. 3b together, we can see that the performance of DQN and DDPG is greatly affected by the coefficient ϱ. The process of choosing this coefficient requires trial and error. However, the proposed CPO approach does not need to determine this coefficient and is effective for reducing the charging cost as well as satisfying the charging constraint. In addition, the MPC method reduces the charging cost by 81.13%. However, it significantly violates the charging constraint by a ratio of 2332.93%. Compared to the MPC method, the CPO method is more effective in handling the charging constraint to meet the user's charging demand. Moreover, the CPO method is model-free and does not needs models to predict the EV departure time and the electricity price.

To further demonstrate the effectiveness of the proposed CPO approach, the charging and discharging schedules of the proposed CPO over 7 consecutive test days (i.e., Jul 7th, 2018 - Jul 16th, 2018) are presented in Fig. 4. Specifically, Fig. 4a shows the hourly electricity prices and EV charging/discharging schedules. Fig. 4b shows the corresponding battery energy of the EV. The areas covered by green in Fig. 4a and Fig. 4b represent the periods of time when the EV is out of home. It can be observed from Fig. 4a that the EV is discharged when the electricity price is high. When the price becomes low, the EV will be charged. When the EV departs home, the battery energy of the EV reaches the charging target. These results validate the effectiveness of the proposed approach in optimizing the real-time EV charging/discharging schedules with the charging demand constraint. Fig. 5 presents

the charging and discharging schedules over another 7 test days (i.e., Dec 2nd, 2018 - Dec 9th, 2018). These test days have different electricity price patterns than those demonstrated in Fig. 4. It can be observed from Fig. 5 that the CPO learns a good policy to charge the EV when the price is low and discharge the EV when the price is high. When the EV departs home, the EV is adequately charged to meet the charging target. The experiment results illustrate that the proposed CPO is adaptive to different electricity price patterns.

V. Conclusion and Discussions

To develop a constrained optimal EV charging/discharging strategy, we formulated the real-time EV charging scheduling problem as a CMDP. In the formulation, we have considered the randomness of the EV's arrival time, departure time and remaining energy, as well as the real-time electricity price. A model-free solution based on SDRL has been proposed. The proposed solution does not require any knowledge about the randomness and the constraint. More importantly, it does not need to manually design a penalty term or tune a penalty coefficient for the constraint. It uses a DNN to directly learn the constrained optimal charging/discharging policy in an end-to-end manner. Experimental results demonstrated that the proposed approach can adequately satisfy the charging constraint and reduce the charging cost compared to the baseline solutions.

In this paper, we take the perspective of EV users and assume they are price-takers. Therefore, we do not consider the impact of EV charging on the price signal. However, when a large fraction of EV users apply the proposed learning algorithm and selfishly shift their charging actions to low price periods, there will be a new peak in the demand side during that periods. This will propel the utility to adjust the electricity price. In this case, the charging action of an EV will result in changes of the future electricity price and in turn affect all EVs' learning. This is a more complicated and significant issue worth discussion from a more systemic perspective. We will leave this problem for future research.

Appendix A
Proof of Corollary 1 and 2

Proof: Let $V^{\pi}(s) = E_{\tau\sim\pi}[\sum_{t=0}^{T}\gamma^t R_t|s_0 = s]$ denote the value function of the formulated CMDP at the state s following the policy π. Given $V^{\pi}(s)$, we define $A^{\pi}(s,a) = R(s,a,s') + \gamma V^{\pi}(s') - V^{\pi}(s)$ as the advantange function of the CMDP at state s taking the action a. Then, we rewrite Eq. (11) by substituting $\delta_f(s,a,s')$ with $A^{\pi}(s,a)$

$$\begin{aligned} L_{\pi,f}(\pi') &= E_{s\sim d^{\pi},a\sim\pi,s'\sim P}\left[\left(\frac{\pi'(a|s)}{\pi(a|s)} - 1\right)\delta_f(s,a,s')\right] \\ &= E_{s\sim d^{\pi},a\sim\pi}\left[\frac{\pi'(a|s)}{\pi(a|s)}A^{\pi}(s,a) - A^{\pi}(s,a)\right] \\ &= E_{s\sim d^{\pi},a\sim\pi}\left[\frac{\pi'(a|s)}{\pi(a|s)}A^{\pi}(s,a)\right] \\ &= \int_s d^{\pi}\int_a \pi(a|s)\frac{\pi'(a|s)}{\pi(a|s)}A^{\pi}(s,a)dads \\ &= \int_s d^{\pi}\int_a \pi'(a|s)A^{\pi}(s,a)dads \\ &= E_{s\sim d^{\pi},a\sim\pi'}\left[A^{\pi}(s,a)\right] \end{aligned} \tag{29}$$

where $E_{s\sim d^{\pi},a\sim\pi}[A^{\pi}(s,a)] = 0$ according to the definition of the advantage function. Substituting (29) into (12), we can derive

$$\begin{aligned} D^{\pm}_{\pi,f}(\pi') = \frac{1}{1-\gamma}E_{\substack{s\sim d^{\pi}\\ a\sim\pi'}} \\ \times\left[A^{\pi}(s,a) \pm \frac{\sqrt{2}\gamma\epsilon^{\pi'}}{1-\gamma}\sqrt{D_{KL}(\pi'||\pi)[s]}\right]. \end{aligned} \tag{30}$$

Since

$$J(\pi') - J(\pi) \geq D^{-}_{\pi,f}(\pi'), \tag{31}$$

we have

$$\begin{aligned} J(\pi') - J(\pi) \geq \frac{1}{1-\gamma}E_{\substack{s\sim d^{\pi}\\ a\sim\pi'}} \\ \times\left[A^{\pi}(s,a) - \frac{\sqrt{2}\gamma\epsilon^{\pi'}}{1-\gamma}\sqrt{D_{KL}(\pi'||\pi)[s]}\right]. \end{aligned} \tag{32}$$

Also, let $V^{\pi}_{C}(s) = E_{\tau\sim\pi}[\sum_{t=0}^{T}\gamma^t c_t|s_0 = s]$ denote the value function with respect to the constraint at the state s following the policy π. Define $A^{\pi}_{C}(s,a) = C(s,a,s') + \gamma V^{\pi}_{C}(s') - V^{\pi}_{C}(s)$ as the advantage function with respect to the constraint at state s taking the action a. Then, Eq. (11) can be similarly rewritten as

$$L_{\pi,f}(\pi') = E_{s\sim d^{\pi},a\sim\pi'}\left[A^{\pi}_{C}(s,a)\right]. \tag{33}$$

Substituting (33) into (12), we can derive

$$D^{\pm}_{\pi,f}(\pi') = \frac{1}{1-\gamma}E_{\substack{s\sim d^{\pi}\\ a\sim\pi'}}\left[A^{\pi}_{C}(s,a) \pm \frac{\sqrt{2}\gamma\epsilon^{\pi'}_{C}}{1-\gamma}\sqrt{D_{KL}(\pi'||\pi)[s]}\right]. \tag{34}$$

Since

$$D^{+}_{\pi,f}(\pi') \geq J_C(\pi') - J_C(\pi), \tag{35}$$

we have

$$\begin{aligned} J_C(\pi') - J_C(\pi) \leq \frac{1}{1-\gamma}E_{\substack{s\sim d^{\pi}\\ a\sim\pi'}} \\ \times\left[A^{\pi}_{C}(s,a) + \frac{\sqrt{2}\gamma\epsilon^{\pi'}_{C}}{1-\gamma}\sqrt{D_{KL}(\pi'||\pi)[s]}\right]. \end{aligned} \tag{36}$$

■

Appendix B
Explanation of Corollary 1 and 2

Let us substitute π and π' in Corollary 1 and 2 with π_{θ^k} and $\pi_{\theta^{k+1}}$, respectively, where π_{θ^k} denotes the charging policy at the kth iteration and $\pi_{\theta^{k+1}}$ denotes the charging

policy at the $k+1$th iteration. Then, we can rewrite Eq. (15) as

$$J(\pi_{k+1}) \geq J(\pi_k) + \frac{1}{1-\gamma} E_{\substack{s\sim d^{\pi^k} \\ a\sim\pi^{k+1}}} \times \left[A^{\pi_k}(s,a) - \frac{\sqrt{2}\gamma\epsilon^{\pi^{k+1}}}{1-\gamma}\sqrt{D_{KL}\left(\pi^{k+1}||\pi^k\right)[s]} \right]. \tag{37}$$

It is noted that the right part of the above inequality is the lower bound of the return $J(\pi_{k+1})$ when we update the policy from π_k to π_{k+1}. If we maximize the lower bound to generate the policy π_{k+1} when at the kth iteration,

$$\begin{aligned}\pi_{k+1} &= \arg\max_{\pi\in\Pi_\theta} J(\pi_k) + \frac{1}{1-\gamma} E_{\substack{s\sim d^{\pi^k} \\ a\sim\pi}} \times \left[A^{\pi_k}(s,a) - \frac{\sqrt{2}\gamma\epsilon^{\pi}}{1-\gamma}\sqrt{D_{KL}\left(\pi||\pi^k\right)[s]} \right] \\ &= \arg\max_{\pi\in\Pi_\theta} E_{\substack{s\sim d^{\pi^k} \\ a\sim\pi}} \left[A^{\pi_k}(s,a) - \frac{\sqrt{2}\gamma\epsilon^{\pi}}{1-\gamma}\sqrt{D_{KL}\left(\pi||\pi^k\right)[s]} \right]\end{aligned} \tag{38}$$

we are guaranteed that an improving policy can be obtained according to Wan *et al.* [16], i.e.,

$$J(\pi_{k+1}) \geq J(\pi_k) \tag{39}$$

Now, we can use the update rule in (38) to improve the policy π_k to maximize the return $J(\pi_k)$. However, we still require the policy π_k to satisfy the charging constraint. To achieve this aim, we use the theoretical result in Corollary 2. Specifically, we rewrite Eq. (16) as

$$J_C(\pi_{k+1}) \leq J_C(\pi_k) + \frac{1}{1-\gamma} E_{\substack{s\sim d^{\pi_k} \\ a\sim\pi_{k+1}}} \times \left[A_C^{\pi_k}(s,a) + \frac{\sqrt{2}\gamma\epsilon_C^{\pi_{k+1}}}{1-\gamma}\sqrt{D_{KL}(\pi_{k+1}||\pi_k)[s]} \right]. \tag{40}$$

The right part of the above inequality is the upper bound of the C-return $J(\pi_{k+1})$ (i.e., the constraint value) when we update the policy from π_k to π_{k+1}. If we constrain the upper bound to be less than or equal to d

$$E_{\substack{s\sim d^{\pi_k} \\ a\sim\pi_{k+1}}} \left[\frac{A_C^{\pi_k}(s,a)}{1-\gamma} + \frac{\sqrt{2}\gamma\epsilon_C^{\pi_{k+1}}}{(1-\gamma)^2}\sqrt{D_{KL}(\pi_{k+1}||\pi_k)[s]} \right] \leq d, \tag{41}$$

we can guarantee that $J(\pi_{k+1})$ is less than or equal to d,

$$J(\pi_{k+1}) \leq d. \tag{42}$$

Combining (38) and (41), we can guarantee that the safe policy update rule in (17) is able to generate a monotonically nondecreasing sequence of policies, $J(\pi_0) \leq \cdots \leq J(\pi_k)$, that satisfy the safety constraint, $J_C(\pi_k) \leq d, k = 1, 2, \ldots, K$.

APPENDIX C
DERIVATION OF THE APPROXIMATE UPDATE RULE

Since π depends on θ, we use $\pi_{\theta k}$ to overload the notation π_k in Eq. (18) to denote the policy at the kth iteration. By taking the first order Taylor series expansion, the objective of the update rule (18) can be approximated around θ^k by

$$\begin{aligned}&E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_\theta}} \left[A^{\pi_{\theta k}}(s,a)\right] \\ &\approx E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_{\theta k}}} \left[A^{\pi_{\theta k}}(s,a)\right] + \nabla_\theta^T E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_\theta}} \left[A^{\pi_{\theta k}}(s,a)\right]\left(\theta-\theta^k\right).\end{aligned} \tag{43}$$

The first term in the expansion $E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_{\theta k}}} [A^{\pi_{\theta k}}(s,a)]$ vanishes according to the definition of the advantage function $A^{\pi_{\theta k}}(s,a)$, so Eq. (43) can be simplified by

$$E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_\theta}} \left[A^{\pi_{\theta k}}(s,a)\right] \approx \nabla_\theta^T E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_\theta}} \left[A^{\pi_{\theta k}}(s,a)\right]\left(\theta-\theta^k\right). \tag{44}$$

Similarly, we can approximate the constraint in (18) by taking the first order Taylor series expansion around θ^k

$$\begin{aligned}&\frac{1}{1-\gamma} E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_\theta}} \left[A_C^{\pi_{\theta k}}(s,a)\right] \\ &\approx E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_{\theta k}}} \left[\frac{A_C^{\pi_{\theta k}}(s,a)}{1-\gamma}\right] + \nabla_\theta^T E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_\theta}} \left[\frac{A_C^{\pi_{\theta k}}(s,a)}{1-\gamma}\right]\left(\theta-\theta^k\right) \\ &= \nabla_\theta^T E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_\theta}} \left[\frac{A_C^{\pi_{\theta k}}(s,a)}{1-\gamma}\right]\left(\theta-\theta^k\right),\end{aligned} \tag{45}$$

Also, by taking the second order Taylor series expansion around θ^k, the KL divergence $\overline{D}_{KL}(\pi_\theta||\pi_{\theta k})$ is approximated by

$$\begin{aligned}\overline{D}_{KL}\left(\pi_\theta||\pi_{\theta k}\right) \approx{}& \overline{D}_{KL}\left(\pi_{\theta k}||\pi_{\theta k}\right) + \nabla_\theta^T \overline{D}_{KL}\left(\pi_\theta||\pi_{\theta k}\right)\left(\theta-\theta^k\right) \\ &+ \frac{1}{2}\left(\theta-\theta^k\right)^T \nabla_{\theta'}^2 \overline{D}_{KL}\left(\pi_\theta||\pi_{\theta k}\right)\left(\theta-\theta^k\right).\end{aligned} \tag{46}$$

The first term in the expansion vanishes because the KL distance between two identical distributions is 0. The second term also vanishes because the KL distance achieves a minimum at $\theta = \theta_k$. Thus $\nabla_\theta \overline{D}_{KL}(\pi_\theta||\pi_{\theta k})|_{\theta=\theta_k} = 0$.

Substitute (44), (45) and (46) into the update rule (18), we can derive an approximation to (18)

$$\begin{aligned}\pi_{\theta k+1} = \arg\max_\theta \;& \nabla_\theta^T E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_\theta}} \left[A^{\pi_{\theta k}}(s,a)\right]\left(\theta-\theta^k\right) \\ s.t.\;\; & J_C\left(\pi_{\theta k}\right) + \nabla_\theta^T E_{\substack{s\sim d^{\pi_{\theta k}} \\ a\sim\pi_\theta}} \times \left[\frac{A_C^{\pi_{\theta k}}(s,a)}{1-\gamma}\right]\left(\theta-\theta^k\right) \leq d, \\ & \frac{1}{2}(\theta-\theta_k)^T \nabla_\theta^2 \overline{D}_{KL}(\pi_\theta||\pi_{\theta k})(\theta-\theta_k) \leq \delta.\end{aligned} \tag{47}$$

APPENDIX D
ESTIMATION OF g, b, H AND c

In practice, it is difficult to calculate the precise values of g, b, H and c. We need to estimate these values by sampling. For $g = \nabla_\theta E_{\substack{s\sim d^{\pi_{\theta^k}} \\ a\sim\pi_\theta}}[A^{\pi_{\theta^k}}(s,a)]$, we can rewrite it as

$$\begin{aligned} g &= \nabla_\theta E_{s\sim d^{\pi_{\theta^k}}}\left[\int_{-\infty}^{\infty}\pi_\theta A^{\pi_{\theta^k}}(s,a)da\right] \\ &= \nabla_\theta E_{s\sim d^{\pi_{\theta^k}}}\left[\int_{-\infty}^{\infty}\frac{\pi_\theta}{\pi_{\theta^k}}\pi_{\theta^k}A^{\pi_{\theta^k}}(s,a)da\right] \\ &= \nabla_\theta E_{s\sim d^{\pi_{\theta^k}}}\left[\frac{\pi_\theta}{\pi_{\theta^k}}E_{a\sim\pi_{\theta^k}}A^{\pi_{\theta^k}}(s,a)\right] \\ &= E_{\substack{s\sim d^{\pi_{\theta^k}} \\ a\sim\pi_{\theta^k}}}\left[\frac{A^{\pi_{\theta^k}}(s,a)}{\pi_{\theta^k}}\right]\nabla_\theta\,\pi_\theta \end{aligned} \tag{48}$$

where the expectation term $E_{\substack{s\sim d^{\pi_{\theta^k}} \\ a\sim\pi_{\theta^k}}}[\frac{A^{\pi_{\theta^k}}(s,a)}{\pi_{\theta^k}}]$ only depends on π_{θ^k}, which is known at the kth iteration; the derivative term $\nabla_\theta\pi_\theta$ is just the gradient of the policy network with respect to the network parameters θ. Therefore, g can be estimated by taking its sample mean when N and M are large,

$$\widehat{g} = \frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M}\frac{A^{\pi_{\theta^k}}(s_n,a_{n,m})}{\pi_{\theta^k}(a_{n,m}|s_n)}\nabla_\theta\,\pi_\theta(a_{n,m}|s_n) \tag{49}$$

where $s_n\sim d^{\pi_{\theta^k}}, n=1,\ldots,N$ are N state samples at the kth iteration; $a_{n,m}, m=1,\ldots,M$ are M action samples at state s_n following the policy π_{θ^k}.

Similarly, we can estimate $b = \nabla_\theta E_{\substack{s\sim d^{\pi_{\theta^k}} \\ a\sim\pi_\theta}}[\frac{A_C^{\pi_{\theta^k}}(s,a)}{1-\gamma}]$ by

$$\widehat{b} = \frac{1}{NM(1-\gamma)}\sum_{n=1}^{N}\sum_{m=1}^{M}\frac{A_C^{\pi_{\theta^k}}(s_n,a_{n,m})}{\pi_{\theta^k}(a_{n,m}|s_n)}\nabla_\theta\,\pi_\theta(a_{n,m}|s_n). \tag{50}$$

For H, we have

$$H = \nabla_\theta^2\overline{D}_{KL}(\pi_\theta||\pi_{\theta^k}) = E_{s\sim d^{\pi^k}}\left[\nabla_\theta^2 D_{KL}(\pi_\theta||\pi_{\theta^k})[s]\right] \tag{51}$$

where

$$\begin{aligned} \nabla_\theta^2 D_{KL}(\pi_\theta||\pi_{\theta^k}) &= \frac{\partial}{\partial\theta}\left(\frac{\partial\pi_\theta}{\partial\theta}\frac{\partial D_{KL}(\pi_\theta||\pi_{\theta^k})}{\partial\pi_\theta}\right) \\ &= \frac{\partial^2\pi_\theta}{\partial\theta^2}\frac{\partial D_{KL}(\pi_\theta||\pi_{\theta^k})}{\partial\theta} \\ &\quad + \left(\frac{\partial\pi_\theta}{\partial\theta}\right)\frac{\partial^2 D_{KL}(\pi_\theta||\pi_{\theta^k})}{\partial\pi_\theta^2}\left(\frac{\partial\pi_\theta}{\partial\theta}\right)^T. \end{aligned} \tag{52}$$

At the point $\theta=\theta^k$, we have

$$\frac{\partial D_{KL}(\pi_\theta||\pi_{\theta^k})}{\partial\theta}|_{\theta=\theta^k} = 0 \tag{53}$$

and

$$\begin{aligned} \frac{\partial^2 D_{KL}(\pi_\theta||\pi_{\theta^k})}{\partial\pi_\theta^2}|_{\theta=\theta^k} &= \begin{bmatrix} \pi_{\theta_1^k}/\pi_{\theta_1}^2 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \pi_{\theta_L^k}/\pi_{\theta_L}^2 \end{bmatrix}_{|\theta=\theta^k} \\ &= \begin{bmatrix} 1/\pi_{\theta_1^k} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1/\pi_{\theta_L^k} \end{bmatrix} \end{aligned} \tag{54}$$

where L is the dimensionality of the action. In our problem, since $L=1$, we can simplify (52) by

$$\nabla_\theta^2 D_{KL}(\pi_\theta||\pi_{\theta^k})|_{\theta=\theta^k} = \left(\frac{\partial\pi_\theta}{\partial\theta}\right)\frac{1}{\pi_{\theta^k}}\left(\frac{\partial\pi_\theta}{\partial\theta}\right)^T. \tag{55}$$

Using (51) and (55), we can estimate H by

$$\widehat{H} = \frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M}\frac{\nabla_\theta\pi_\theta(a_{n,m}|s_n)\,\nabla_\theta^T\,\pi_\theta(a_{n,m}|s_n)}{\pi_{\theta^k}(a_{n,m}|s_n)}. \tag{56}$$

For $c = J_C(\pi_{\theta^k}) - d$, since d is a known value and $J_C(\pi_{\theta^k})$ is estimated by the sample mean of C-returns

$$J_C(\pi_{\theta^k}) \approx \frac{1}{D}\sum_{d=1}^{D}\left[\sum_{t=0}^{T}\gamma^t c_t\right], \tau\sim\pi_{\theta^k},$$

thus we can estimate c by

$$\widehat{c} = \frac{1}{D}\sum_{d=1}^{D}\sum_{t=0}^{T}\gamma^t c_t - d.$$

REFERENCES

[1] W. Tang, S. Bi, and Y. J. Zhang, "Online charging scheduling algorithms of electric vehicles in smart grid: An overview," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 76–83, Dec. 2016.

[2] R. Moghaddass, O. A. Mohammed, E. Skordilis, and S. Asfour, "Smart control of fleets of electric vehicles in smart and connected communities," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6883–6897, Nov. 2019.

[3] *The Egallon: How Much Cheaper Is it to Drive on Electricity?* Accessed: May 17, 2019. [Online]. Available: https://energy.gov/articles/egallon-howmuch-cheaper-it-drive-electricity

[4] *Global EV Outlook 2018*, OECD/IEA, Int. Energy Agency, Paris, France, May 2018. [Online]. Available: https://www.iea.org/gevo2018/

[5] F. Rassaei, W.-S. Soh, and K.-C. Chua, "Demand response for residential electric vehicles with random usage patterns in smart grids," *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1367–1376, Oct. 2015.

[6] T. Namerikawa, N. Okubo, R. Sato, Y. Okawa, and M. Ono, "Real-time pricing mechanism for electricity market with built-in incentive for participation," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2714–2724, Nov. 2015.

[7] H. S. V. S. K. Nunna, S. Battula, S. Doolla, and D. Srinivasan, "Energy management in smart distribution systems with vehicle-to-grid integrated microgrids," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4004–4016, Sep. 2018.

[8] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[9] D. Silver, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[10] Z. Wan, C. Jiang, M. Fahad, Z. Ni, Y. Guo, and H. He, "Robot-assisted pedestrian regulation based on deep reinforcement learning," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2018.2878977.

[11] Z. Wen, D. O'Neill, and H. Maei, "Optimal demand response using device-based reinforcement learning," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2312–2324, Sep. 2015.

[12] S. Vandael, B. Claessens, D. Ernst, T. Holvoet, and G. Deconinck, "Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1795–1805, Jul. 2015.
[13] A. Chiş, J. Lundén, and V. Koivunen, "Reinforcement learning-based plug-in electric vehicle charging with forecasted price," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 3674–3684, May 2017.
[14] Z. Wan and H. He, "AnswerNet: Learning to answer questions," *IEEE Trans. Big Data*, to be published, doi: 10.1109/TBDATA.2018.2884486.
[15] Z. Wan, H. He, and B. Tang, "A generative model for sparse hyperparameter determination," *IEEE Trans. Big Data*, vol. 4, no. 1, pp. 2–10, Mar. 2018.
[16] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2019.
[17] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, no. 10, 2017, pp. 22–31.
[18] R. Li, Q. Wu, and S. S. Oren, "Distribution locational marginal pricing for optimal electric vehicle charging management," *IEEE Trans. Power Syst.*, vol. 29, no. 1, pp. 203–211, Jan. 2014.
[19] S. I. Vagropoulos, G. A. Balaskas, and A. G. Bakirtzis, "An investigation of plug-in electric vehicle charging impact on power systems scheduling and energy costs," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 1902–1912, May 2017.
[20] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 1889–1897.
[21] *Day-Ahead and Historical RTP/HSS Prices*, Ameren, St. Louis, MO, USA, Jan. 2019. [Online]. Available: https://www.ameren.com/account/retail-energy
[22] A. Mohamed, V. Salehi, T. Ma, and O. Mohammed, "Real-time energy management algorithm for plug-in hybrid electric vehicle charging parks involving sustainable energy," *IEEE Trans. Sustain. Energy*, vol. 5, no. 2, pp. 577–586, Apr. 2014.
[23] R. Wang, P. Wang, and G. Xiao, "Two-stage mechanism for massive electric vehicle charging involving renewable energy," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4159–4171, Jun. 2016.
[24] K. Chaudhari, A. Ukil, K. N. Kumar, U. Manandhar, and S. K. Kollimalla, "Hybrid optimization for economic deployment of ESS in PV-integrated EV charging stations," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 106–116, Jan. 2018.
[25] L. Yao, W. H. Lim, and T. S. Tsai, "A real-time charging scheme for demand response in electric vehicle parking station," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 52–62, Jan. 2017.
[26] A. Gleixner *et al.* (Jul. 2018). *The SCIP Optimization Suite 6.0*. [Online]. Available: https://scip.zib.de/

Hepeng Li (S'19) received the B.S. degree in information and computing science and the M.S. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2009 and 2012, respectively. He is currently pursuing the Ph.D. degree with the School of Electrical, Computer and Biomedical Engineering, University of Rhode Island, South Kingstown, RI, USA. From 2014 to 2019, he was an Assistant Professor with the Institute of Automation, Chinese Academy of Sciences, Shenyang. His research interests include smart gird, microgrids, demand response, deep neural networks, and deep reinforcement learning.

Zhiqiang Wan (S'16) received the B.S. degree from the Harbin Institute of Technology, Harbin, China, in 2012, and the M.S. degree from the School of Electrical and Electronics Engineering, Huazhong University of Science and Technology in 2015. He is currently pursuing the Ph.D. degree with the School of Electrical, Computer and Biomedical Engineering, University of Rhode Island (URI), South Kingstown, RI, USA. His current research interests include deep learning, deep reinforcement learning, and cyber-physical system, with a particular interest in smart grid applications. He was a recipient of the URI Graduate Student Research & Scholarship Excellence Award in the Life Sciences, Physical Sciences, and Engineering in 2019, the Best Paper Award in the IEEE Power & Energy Society General Meeting in 2018, and the Best Paper Award in the IEEE 11th International Conference on Power Electronics and Drive Systems in 2015.

Haibo He (SM'11–F'18) received the B.S. and M.S. degrees in electrical engineering from the Huazhong University of Science and Technology, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Ohio University in 2006. He is currently the Robert Haas Endowed Chair Professor with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island. His current research interests include computational intelligence, machine learning, data mining, and various applications. He received the IEEE International Conference on Communications Best Paper Award in 2014, the IEEE CIS Outstanding Early Career Award in 2014, and the National Science Foundation CAREER Award in 2011. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He was the General Chair of the IEEE Symposium Series on Computational Intelligence (SSCI 2014).