Safe Off-policy Deep Reinforcement Learning Algorithm for Volt-VAR Control in Power Distribution Systems

Wei Wang, *Student Member, IEEE,* Nanpeng Yu, *Senior Member, IEEE,* Yuanqi Gao, *Student Member, IEEE,* Jie Shi, *Student Member, IEEE*

***Abstract*—Volt-VAR control is critical to keeping distribution network voltages within allowable range, minimizing losses, and reducing wear and tear of voltage regulating devices. To deal with incomplete and inaccurate distribution network models, we propose a safe off-policy deep reinforcement learning algorithm to solve Volt-VAR control problems in a model-free manner. The Volt-VAR control problem is formulated as a constrained Markov decision process with discrete action space, and solved by our proposed constrained soft actor-critic algorithm. Our proposed reinforcement learning algorithm achieves scalability, sample efficiency, and constraint satisfaction by synergistically combining the merits of the maximum-entropy framework, the method of multiplier, a device-decoupled neural network structure, and an ordinal encoding scheme. Comprehensive numerical studies with the IEEE distribution test feeders show that our proposed algorithm outperforms the existing reinforcement learning algorithms and conventional optimization-based approaches on a large feeder.**

***Index Terms*—Deep reinforcement learning, model-free, off-policy, safe reinforcement learning, Volt-VAR control.**

I. Introduction

TO tackle the challenge of managing distribution system-wide voltage levels and reactive power flows, Volt-VAR control (VVC) has been developed and integrated into the distribution management system. VVC determines the best set of control actions for all voltage regulating and VAR control devices (voltage regulators, on-load tap changers, and switchable capacitor banks) to reduce system losses and equipment operating costs without violating operation constraints such as voltage limits and line flow limits.

The existing VVC algorithms deployed by electric utilities mainly adopt the physical model-based control approach, which relies heavily on accurate knowledge of distribution grid topologies and parameters. However, it is difficult for regional electric utilities to maintain reliable network models [1], [2], which often involve millions of buses in the primary and secondary feeders [3]. To cope with incomplete models, one could learn which VVC actions yield the most reward by trying them. Moreover, the model-based control approaches are not always scalable and may not be applicable in real-time control environment. It has been shown that the deep reinforcement learning (DRL) approach could overcome this problem in emergency system control [4], [5]. In this paper, we propose a safe off-policy DRL algorithm to learn and execute VVC actions in a model-free manner.

The majority of the existing work on VVC adopt a physical model-based optimization/control approach. Due to space limitation, we focus on summarizing recent advancements of VVC technology, which can be separated into three groups. The first group of literature formulates VVC as deterministic optimization problems. The VVC problem is extended to consider voltage-dependent loads [6] and formulated as a mixed-integer quadratically constrained programming problem. A power electronic device, called soft open point [7], is introduced to achieve real-time VVC together with conventional voltage regulation devices. The coordinated control problem is formulated as a mixed-integer second-order conic programming problem. The VVC problem is formulated as a non-cooperative mixed strategy game [8], which considers flexible loads, electric vehicles, and renewable energy sources. The limit on the number of switching operations of voltage regulating devices is considered in the VVC [9], which is formulated as a mixed-integer nonlinear programming problem.

The second group of literature explicitly incorporates the uncertainties of DERs in the VVC problem formulation. A dual time-scale coordination scheme for slow and fast controlling devices is proposed for the VVC problem, which is solved with stochastic [10] and robust [11] optimization algorithms. The model predictive control (MPC) based VVC algorithms are proposed to reduce network losses [12], voltage deviations and excessive wear and tear of voltage regulating devices [13].

To address the algorithm scalability problem and the communication delay issue of the centralized optimization and control approach for VVC, the third group of literature develops non-centralized control schemes, which can be further divided into three subgroups, local VVC algorithms, distributed VVC algorithms, and decentralized VCC algorithms [14]. The local VVC algorithms use only locally available information such as bus voltages to design control strategies. Fully decentralized disturbance-feedback controller [15], gradient-projection algorithm [16], and asynchronous gradient-project algorithm are developed [17] for local voltage controls. The distributed VVC algorithms allow neighboring agents to communicate and share information to cooperatively reach global objectives of VVC. Distributed algorithms such as the alternating direction method of multipliers [18], the

This work was supported in part by the Department of Energy (DOE) under award DE-OE000840 and California Energy Commission (CEC) under award EPC-15-090. (Corresponding author: Nanpeng Yu).

The authors are with the Department of Electrical and Computer Engineering at University of California, Riverside. (email: wwang031@ucr.edu; nyu@ece.ucr.edu; ygao024@ucr.edu; jshi005@ucr.edu).

dual decomposition method [19], the integral-control-like update scheme [20], and the local optimization and consensus approach [21] are developed to solve VVC problems. The decentralized algorithms are developed with ϵ-decomposition in [22], [23], where centralized control is only needed within the isolated sub-areas.

To remove the dependency on complete and accurate distribution network topology and parameter information, a few researchers have developed reinforcement learning (RL) based algorithms for VVC. The tabular Q-learning algorithm [24] is used to learn the setting of control variables which satisfy operation constraints in power systems. The tabular Q-learning method with the global reward recovered from the consensus-based algorithm [25] is proposed to solve the optimal reactive power dispatch problem. Radial basis functions are used to approximate Q-function in [26] to find the optimal tap settings of the voltage regulation devices. In the existing RL-based algorithms, the VVC problems are always modeled as Markov decision process (MDP) and solved with Q-learning algorithm, which is a commonly used action-value method in RL. The action-value methods [27] learn to approximate the action-value functions and then select actions based on the estimated action-value functions and the ϵ-greedy algorithm [28].

In this paper, we propose a safe off-policy deep reinforcement learning algorithm to solve the VVC problem with voltage regulating devices. Unlike the existing RL-based VVC algorithms, we formulate the VVC problems as a constrained MDP (CMDP) and propose a novel policy gradient method, called constrained soft actor-critic (CSAC), to solve the CMDP. In contrast to action-value methods, policy gradient methods [29], [30] learn a parameterized control policy that directly selects actions. For VVC problems, it is much simpler to approximate the control policy functions than to approximate the action-value functions for action taking. This is one of the major advantages of adopting policy parameterization, which will be shown in the numerical study.

Compared to the existing RL-based VVC algorithms, our proposed CSAC algorithm is safe, scalable, and, sample efficient. The main contributions of this paper and the technical advancements are summarized as follows:

- Instead of penalizing constraints violation in the reward function of MDP, we propose a CMDP formulation for the VVC problem, which explicitly models the physical operation constraints. By synergistically combining the merits of the method of multipliers and soft actor-critic (SAC) [31] algorithm, our proposed CSAC algorithm can better satisfy the operation constraints in power distribution systems.
- Compared to tabular Q-learning and deep Q-network (DQN) [32], [33], our proposed CSAC algorithm has significantly improved scalability. By designing the policy neural network with a device-decoupled structure, the number of parameters in our proposed method increases linearly with the number of voltage regulating devices. On the other hand, in Q-learning based approach, the number of parameters increases exponentially with the number of voltage regulating devices.
- Our proposed CSAC is an off-policy method, which is more sample efficient than state-of-the-art DRL algorithms for CMDP such as constrained policy optimization (CPO) [34], [35]. This is because, our proposed method can effectively reuse historical operational data for training purpose. Furthermore, by using an ordinal network structure to encode the natural ordering between discrete actions of voltage regulating devices, the inductive bias can be introduced to further accelerate the learning process.
- In contrast to physical model-based VVC algorithms, our proposed DRL approach is model-free and does not rely on complete and accurate distribution network topology models or parameters.

The remainder of the paper is organized as follows. Section II provides the formulation of the VVC problem as a CMDP. The proposed safe off-policy DRL algorithm is presented in Section III. Section IV shows the results of our numerical study. Section VI states the conclusions.

II. Problem Formulation

In this section, we first introduce the preliminaries for CMDP and then formulate the VVC problem as a CMDP.

A. Preliminaries of Constrained Markov Decision Process

As a formalization of sequential decision making, CMDP is defined by a tuple of a state space $\mathcal{S}$, an action space $\mathcal{A}$, a reward function R, a cost function R^c, a transition probability function Pr, and a discount factor $\gamma \in (0, 1)$.

In a CMDP, a learner and decision maker, also called an agent, interacts with the environment at each of a sequence of discrete time steps, $t = 0, 1, 2, 3, \ldots, T$. At each time step t, the agent observes the state of the environment $\boldsymbol{s}_t \in \mathcal{S}$ and selects an action $\boldsymbol{a}_t \in \mathcal{A}$. One time step later, the agent receives a numerical reward $R(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) \subset \mathbb{R}$ and a numerical cost $R^c(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) \subset \mathbb{R}$. The state of the environment becomes $\boldsymbol{s}_{t+1}$ according to the transition probability function $Pr(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$.

The goal of the agent is to find a control policy π that maximizes the expected discounted return with respect to reward function J subject to a budget constraint for the expected discounted return with respect to cost function J^c:

$$\max_{\pi} J(\pi) \quad s.t. \quad J^c(\pi) \leq \overline{J} \tag{1}$$

where π is a mapping from a state space $\mathcal{S}$ to a action space $\mathcal{A}$ for a deterministic policy or a mapping from states to probabilities of selecting different actions for a probabilistic policy. The expected discounted return of policy π with respect to the reward is defined as: $J(\pi) = \underset{\tau\sim\pi}{E}[\sum_{t=0}^{T} \gamma^t R_t]$, where τ is a trajectory or sequence of states and actions, $\{\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{s}_1, \boldsymbol{a}_1, ..., \boldsymbol{s}_{T-1}, \boldsymbol{a}_{T-1}, \boldsymbol{s}_t\}$. R_t is the short name for $R(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$. Similarly, the expected discounted return of policy π with respect to cost function is defined as $J^c(\pi) = \underset{\tau\sim\pi}{E}[\sum_{t=0}^{T} \gamma^t R_t^c]$, where R_t^c is $R^c(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$ for short.

Finally, we define two important value functions, state-value function $V^\pi(\boldsymbol{s})$ and action-value function $Q^\pi(\boldsymbol{s},\boldsymbol{a})$, as follows:

$$V^\pi(\boldsymbol{s}) = \underset{\tau\sim\pi}{E}[\sum_{t=0}^{T}\gamma^t R_t|\boldsymbol{s}_0=\boldsymbol{s}] \tag{2}$$

$$Q^\pi(\boldsymbol{s},\boldsymbol{a}) = \underset{\tau\sim\pi}{E}[\sum_{t=0}^{T}\gamma^t R_t|\boldsymbol{s}_0=\boldsymbol{s},\boldsymbol{a}_0=\boldsymbol{a}] \tag{3}$$

$V^\pi(\boldsymbol{s})$ represents the expected discounted return starting from state $\boldsymbol{s}$ and taking actions following policy π thereafter. $Q^\pi(\boldsymbol{s},\boldsymbol{a})$ represents the expected discounted return starting from state $\boldsymbol{s}$, taking action $\boldsymbol{a}$, and thereafter following policy π. The value functions satisfy the Bellman equations:

$$V^\pi(\boldsymbol{s}_t) = \underset{\substack{a_t\sim\pi\\ \boldsymbol{s}_{t+1}\sim Pr}}{E}[R_t+\gamma V^\pi(\boldsymbol{s}_{t+1})] \tag{4}$$

$$Q^\pi(\boldsymbol{s}_t,\boldsymbol{a}_t) = \underset{\substack{\boldsymbol{a}_{t+1}\sim\pi\\ \boldsymbol{s}_{t+1}\sim Pr}}{E}\Big[R_t+\gamma Q^\pi(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1})\Big] \tag{5}$$

B. Formulating VVC Problem as a CMDP

In the VVC problem, the distribution system operator or controller is treated as the agent who interacts with the distribution grid. In this paper, the primary controllable devices for the VVC task are selected to be voltage regulators, on-load tap changers, and switchable capacitor banks. The state of the environment is defined as $\boldsymbol{s}=(\boldsymbol{P},\boldsymbol{Q},\boldsymbol{Tap},t)$. $\boldsymbol{P}$ and $\boldsymbol{Q}$ are the vectors of nodal real and reactive power injections. $\boldsymbol{Tap}$ is the vector of the current tap/on-off positions of controllable devices. The action taken by a VVC agent at each time step is changing the tap/on-off positions of controllable devices to $\boldsymbol{Tap}'$. The size of the action space is $\Pi_{i=1}^{N_c}|\mathcal{A}_i|$, where N_c is the number of controllable devices and $|\mathcal{A}_i|$ denotes the number of tap/on-off positions of device i.

The VVC agent aims at reducing the distribution network losses and the operating costs of the controllable devices. Thus, the reward function R_t of the VVC agent can be defined as the negative of the total operational costs, which includes the cost of real power losses and the device switching cost:

$$R_t = -\Big[C_e P_{loss}(t) + \sum_{j=1}^{N_c} C_j^T|Tap_j(t+1)-Tap_j(t)|\Big] \tag{6}$$

The switching cost of a device is calculated as the product of the absolute change in tap positions between consecutive time steps and the per tap position change cost C_j^T for device j. C_e and $P_{loss}(t)$ denote the electricity price and the total real power loss at time step t respectively. The total real power loss is defined as the summation of real power losses of all lines and devices in the distribution network.

To maintain nodal voltage profiles within a desirable range, the cost function is chosen as the number of voltage constraint violations across all the nodes:

$$R_t^c = \sum_{i=1}^{N}[\mathbb{1}(|V_i^{t+1}|>\overline{V}) + \mathbb{1}(|V_i^{t+1}|<\underline{V})] \tag{7}$$

where $\mathbb{1}(\cdot)$ is the indicator function. V_i^{t+1} is the voltage of node i at hour $t+1$; $\overline{V}$ and $\underline{V}$ are the upper and lower limits for voltage magnitudes. N is the total number of nodes.

By evaluating the feedback in the form of rewards and costs defined above via past and/or future interactions with the physical environment, the VVC agent tries to learn a control policy that minimizes the total operational cost while satisfying the voltage constraints.

III. Safe Off-policy Deep RL Algorithm

In this section, we develop an innovative DRL algorithm named constrained soft actor-critic (CSAC) to solve the VVC problem, which is formulated as a CMDP. A suitable RL algorithm for solving the VVC problem should be sample efficient, scalable, and safe to implement in the real world.

Sample efficiency: Unlike the domain of computer games, we can not repeatedly generate a tremendous amount of operation experiences for VVC in real world distribution feeders with low cost. Thus, it is crucial for us to develop off-policy RL algorithms, where the learned control policy (target policy) and the policy that generated control behaviors (behavior policy) are different. Being able to reuse the historical operational experiences, the off-policy RL algorithms are much more sample efficient than the on-policy ones.

Scalability: In a VVC problem, the network loss is determined by the tap positions of all controllable devices together. The number of feasible control actions increases exponentially with the number of controllable devices. Thus, in order to solve a large-scale VVC problem, it is important to learn a control policy whose number of parameters increases approximately linearly with the number of controllable devices.

Constraint satisfaction: In RL, agents are often given complete freedom to learn a control policy by trial and error. However, in a real-world VVC problem, this is unacceptable. Certain exploratory control actions may lead to significant voltage violations in the distribution network causing equipment damage and undermining the reliability of the network. Thus, we want to develop a RL algorithm, which can achieve near constraint satisfaction at all times.

In the following subsections, we first introduce the actor-critic method, which is a widely used policy gradient method. Next, we present the state-of-the-art maximum-entropy based off-policy RL algorithm, soft actor-critic (SAC). We then propose an innovative off-policy RL algorithm called CSAC to solve the VVC problem. This is followed by a presentation of the detailed algorithm design for CSAC. At last, we derive the policy gradient for discrete actions and describe the device-decoupled policy network structure and ordinal encoding for discrete actions.

A. Actor-Critic Method

The basic policy gradient method is an actor-only method, where the actor refers to the policy function. Actor-only methods typically learn parameters for the approximated policy function based on episodic gains from Monte-Carlo sample trajectories. This often leads to high variance and slow learning [28]. To overcome these shortcomings, the actor-critic method

is proposed to update policy function parameters based on the approximated value function that is a synonym for the critic. The iterative framework for a typical actor-critic method is shown in Algorithm 1. At each iteration, the actor first generates samples by taking actions according to the current policy. Then, the critic evaluates the quality of the current policy by adjusting the value function estimates based on the temporal difference [28] according to (5). At last, the actor is updated by using the information from the critic.

Algorithm 1 Actor-Critic Algorithm

1: Initialize policy and value function parameters
2: **repeat**
3: Generate samples by taking actions according to the current policy
4: Update value function parameters according to (5)
5: Update policy parameters based on value function
6: **until** converge

B. Soft Actor-Critic

The commonly used actor-critic algorithms such as PPO [36] and A3C [37] are notoriously sample inefficient, because they require new samples to be generated according to the latest policy at each gradient step. Although off-policy policy gradient algorithms such as DDPG [29] were introduced to improve sample efficiency, they are often brittle with respect to their hyperparameters. To address these challenges, the off-policy maximum-entropy deep RL algorithm, SAC [31], is developed to provide a robust and sample-efficient learning, which achieves the state-of-the-art performance.

The SAC is built on the maximum-entropy RL framework [38], [39], which maximizes not only the expected return but also the entropy of the policy. The entropy for a probabilistic policy at state $\boldsymbol{s}_t$ is defined as $H(\pi(\cdot|\boldsymbol{s}_t)) = -\sum_{\boldsymbol{a}} \pi(\boldsymbol{a}|\boldsymbol{s}_t) \ln \pi(a|\boldsymbol{s}_t)$.

In the maximum-entropy RL framework, we typically work with the regularized value functions [40] defined as:

$$V_h^{\pi}(\boldsymbol{s}) = \underset{\tau\sim\pi}{E}\Big[\sum_{t=0}^{T}\gamma^t\big(R_t + \alpha H(\pi(\cdot|\boldsymbol{s}_t))\big)\big|\boldsymbol{s}_0 = s\Big] \tag{8}$$

$$Q_h^{\pi}(\boldsymbol{s},\boldsymbol{a}) = \underset{\tau\sim\pi}{E}\Big[\sum_{t=0}^{T}\gamma^t R_t + \alpha\sum_{t=1}^{T}\gamma^t H(\pi(\cdot|\boldsymbol{s}_t))\big|\boldsymbol{s}_0 = s, \boldsymbol{a}_0 = a\Big] \tag{9}$$

The corresponding entropy-regularized Bellman equations are:

$$V_h^{\pi}(\boldsymbol{s}_t) = \underset{\substack{a_t\sim\pi\\ \boldsymbol{s}_{t+1}\sim Pr}}{E}\Big[R_t + \alpha H(\pi(\cdot|\boldsymbol{s}_t)) + \gamma V_h^{\pi}(\boldsymbol{s}_{t+1})\Big] \tag{10}$$

$$Q_h^{\pi}(\boldsymbol{s}_t,\boldsymbol{a}_t) = \underset{\substack{\boldsymbol{a}_{t+1}\sim\pi\\ \boldsymbol{s}_{t+1}\sim Pr}}{E}\Big[R_t + \gamma\big(Q_h^{\pi}(\boldsymbol{s}_{t+1},\boldsymbol{a}_{t+1}) + \alpha H(\pi(\cdot|\boldsymbol{s}_{t+1}))\big)\Big] \tag{11}$$

The two regularized value functions have the following relationship:

$$V_h^{\pi}(\boldsymbol{s}_t) = \underset{\boldsymbol{a}_t\sim\pi}{E}[Q_h^{\pi}(\boldsymbol{s}_t,\boldsymbol{a}_t)] + \alpha H(\pi(\cdot|\boldsymbol{s}_t)) \tag{12}$$

Equation (12) allows us to derive the closed-form solution [40] of the policy $\pi^{\dagger}(\cdot|s) = arg\max_{\pi\in\Delta}\{V_h^{\pi}(\boldsymbol{s})\}$, where $\Delta = \{\pi|\pi \geq 0, \mathbf{1}\cdot\pi = 1\}$, as:

$$\pi^{\dagger}(\cdot|\boldsymbol{s}) = \frac{e^{Q_h^{\pi}(\boldsymbol{s},\cdot)/\alpha}}{\sum_{\boldsymbol{a}} e^{Q_h^{\pi}(\boldsymbol{s},\boldsymbol{a})/\alpha}} \tag{13}$$

When Q_h^{π} converges to Q_h^{*}, the optimal policy $\pi^{*}(\cdot|\boldsymbol{s})$ also achieves optimal value $V_h^{*}(\boldsymbol{s})$ for all states $\boldsymbol{s}$. By using the closed-form solution, the updating schema of Q-function could be realized in an off-policy fashion.

Algorithm 2 Soft Actor-Critic

1: Initialize policy and regularized value function parameters
2: **repeat**
3: Sample from data buffer
4: Update parameters of value functions according to (11)
5: Update policy parameters according to (13)
6: **until** converge

The overall framework of SAC is summarized in Algorithm 2. The implementation details such as the clipped double-Q learning [41], the baseline value function [28], and the delayed update of value function [31] are omitted here.

C. Constrained Soft Actor-Critic

Although SAC has been successfully demonstrated on a range of challenging control tasks, it is designed to solve MDPs and cannot handle CMDPs with physical constraints. If one simply augments the reward with the product of a fixed penalty factor and constraint violation, then the learned policy will be either too conservative or infeasible. In this subsection, we propose CSAC by extending SAC algorithm to satisfy the operational constraints in CMDPs.

The goal of the SAC algorithm is to find an optimal policy, which maximizes the regularized state-value function, $\max_{\pi} \underset{\boldsymbol{s}\sim D}{E}[V_h^{\pi}(\boldsymbol{s})]$, where D is the historical operation data buffer, i.e., the set of experience tuple $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, R_t, R_t^c)$.

Moreover, in real-world control problems, it is necessary to enforce operational constraints. For the VVC problem, we need to limit the number of total voltage constraint violations at each time step, i.e., $R_t^c \leq \overline{R^c}$. R_t^c is defined in (7), and $\overline{R^c}$ is the upper bound. For a finite horizon CMDP, the corresponding limit $\overline{V^c}$ for the state-value function associated with the operation constraint can be set as $V^{c,\pi}(\boldsymbol{s}) \leq \overline{V^c} = (1-\gamma^T)/(1-\gamma)\overline{R^c}$, where T is the episode length. Note that other types of operational constraints can be enforced in a similar manner.

Within the maximum-entropy RL framework, the optimal policy of CMDP can be obtained by solving:

$$\max_{\pi} \underset{\boldsymbol{s}\sim D}{E}[V_h^{\pi}(\boldsymbol{s})], \quad s.t \quad \underset{\boldsymbol{s}\sim D}{E}[V^{c,\pi}(\boldsymbol{s})] \leq \overline{V^c} \tag{14}$$

The Lagrange function of the constrained optimization problem can be written as:

$$\begin{aligned}\mathcal{L}(\pi,\lambda) &= \underset{\boldsymbol{s}\sim D}{E}[V_h^{\pi}(\boldsymbol{s})] + \lambda(\overline{V^c} - \underset{\boldsymbol{s}\sim D}{E}[V^{c,\pi}(\boldsymbol{s})]) \\ &= \underset{\boldsymbol{s}\sim D}{E}[V_h^{l,\pi}(\boldsymbol{s})] + \lambda\overline{V^c}\end{aligned} \tag{15}$$

where

$$V_h^{l,\pi}(\boldsymbol{s}) = \underset{\tau\sim\pi}{E}\Big[\sum_{t=0}^{T}\gamma^t\big(R(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1}) - \lambda R^c(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1}) + \alpha H(\pi(\cdot|\boldsymbol{s}_t))\big|\boldsymbol{s}_0=\boldsymbol{s}\Big] \tag{16}$$

The method of multipliers can be used to solve the constrained optimization problem. At k-th iteration , given a multiplier $\lambda^k \geq 0$, we can maximize $\mathcal{L}(\cdot,\lambda^k)$, over policy domain thereby obtaining a policy π^k. We then set

$$\lambda^{k+1} = [\lambda^k - \delta_\lambda \nabla_\lambda \mathcal{L}]^+ = [\lambda^k + \delta_\lambda(\underset{\boldsymbol{s}\sim D}{E}[V^{c,\pi_k}(\boldsymbol{s})] - \overline{V^c})]^+ \tag{17}$$

and repeat the process. δ_λ is the step size for the λ update process. $[]^+$ is the projection to non-negative real numbers.

With a small α and $H(\pi)\approx 0$ at convergence, $V_h^{c,\pi}(\boldsymbol{s}) \approx V^{c,\pi}(\boldsymbol{s})$. To have consistent forms of value functions, the update of the Lagrange multiplier can be redesigned as:

$$\lambda^{k+1} = [\lambda^k + \delta_\lambda(\underset{\boldsymbol{s}\sim D}{E}[V_h^{c,\pi_k}(\boldsymbol{s})] - \overline{V^c})]^+ \tag{18}$$

where $V_h^{c,\pi_k}(\boldsymbol{s})$ is the state-value function associated with the operation constraint at k-th iteration.

It has been shown that the iterative approach for updating the parameters of control policy and Lagrange multiplier will guarantee the convergence to a local optimal and feasible solution when the following three assumptions hold [42], [43]. First, $V_h^{\pi}(\boldsymbol{s})$ is bounded for all policies $\pi \in \Pi$. Second, every local minima of $J^c(\pi)$ is a feasible solution. Third, $\sum_{k=0}^{\infty}\delta_\theta = \sum_{k=0}^{\infty}\delta_\lambda = \infty$, $\sum_{k=0}^{\infty}\delta_\theta^2 + \sum_{k=0}^{\infty}\delta_\lambda^2 < \infty$, and $\lim_{k\to\infty}\delta_\lambda/\delta_\theta = 0$. δ_θ is the step size for updating the parameters θ of the policy neural network.

Note that for finite episodic cases, δ_λ can be set to be smaller than δ_θ in practice. If the local optimal solution is not feasible, then the algorithm can be restarted with a larger initial value for λ.

D. Algorithm Design for CSAC

The proposed CSAC is an off-policy RL algorithm, which allows the offline training of control policy in an iterative manner. The overall framework of the CSAC is summarized in Algorithm 3. In each iteration, we first perform stochastic gradient descent to update the parameters of neural networks, which approximate the value functions and policy function. Then, we update the Lagrange multiplier of the constrained optimization problem as shown in (18).

Two sets of neural networks are used to approximate the action-value functions Q_ψ and state-value functions V_ϕ. The first set of value functions, parameterized with ψ^l and ϕ^l, are associated with the value functions in the Lagrange function (15). The second set of value functions, parameterized with ψ^c and ϕ^c, are associated with the constraint. The policy function is approximated by a neural network π_θ parameterized by θ.

The parameters of the action-value neural networks, Q_ψ, are updated by minimizing the mean-square-error (MSE), $1/|B|\sum_B(Q_\psi - \hat{Q})^2$, where B is a randomly selected mini-batch of samples, i.e., a set of transition tuples $\{(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1},R_t,R_t^c)\}$. $|B|$ denotes the size of the mini-batch. The training target $\hat{Q}$ is calculated as $\hat{Q}(\boldsymbol{s}_t,\boldsymbol{a}_t) = r_t + \gamma V_\psi(\boldsymbol{s}_{t+1})$, where r_t is $R_t - \lambda R_t^c$ for the neural network associated with the Lagrange function and R_t^c for the neural network associated with the constraint. Similarly, the state-value networks, V_ϕ, are updated by minimizing the MSE, $1/|B|\sum_B(V_\phi - \hat{V})^2$, where the target $\hat{V}(\boldsymbol{s}_t) = Q_\psi(\boldsymbol{s}_t,\boldsymbol{a}_t) - \alpha\ln\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)$. The parameters of the policy neural network is updated by minimizing the loss,

$$\frac{1}{|B|}\sum_B \ln\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t)(\alpha\ln\pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t) - Q_{\psi^l}(\boldsymbol{s}_t,\hat{\boldsymbol{a}}_t) + V_{\phi^l}(\boldsymbol{s}_t)) \tag{19}$$

where $\hat{\boldsymbol{a}}_t$ is the sampled action from $\pi_\theta(\cdot|\boldsymbol{s}_t)$. The derivation for the policy gradient is provided in the subsection III-E.

Algorithm 3 CSAC Algorithm

```
Initialize network parameters and Lagrange multiplier λ
repeat
    for each sample step do
        a_t ~ π(·|s_t)
        D ← D ∪ (s_t, a_t, s_{t+1}, R_t, R_t^c)
    end for
    for each gradient step with sample batch B do
        Update action value networks Q_ψ
        Update state value networks V_φ
        Update policy network π_θ
        λ ← [λ + δ_λ Σ_B(V_{φ^c} − V̄^c)/|B|]^+
    end for
until converge
```

The neural networks approximating V and Q functions use the state vector $\boldsymbol{s}$ and the state action pair $\boldsymbol{s},\boldsymbol{a}$ as inputs, where $\boldsymbol{a}$ is treated as a vector of ordinal variables. The outputs of these two networks are the corresponding target state and action values. The policy network has a special design, which will be described in subsection III-F.

In order to stabilize the training process, the delayed update of value function [31] is adopted in our algorithm. The training labels for Q networks are modified as $\hat{Q}(\boldsymbol{s}_t,\boldsymbol{a}_t) = r_t + \gamma V_{\psi_{targ}}(\boldsymbol{s}_{t+1})$, where $V_{\psi_{targ}}$ are the extra copies of V networks, whose parameter ψ_{targ} updates are delayed at each gradient step by $\phi_{targ} = (1-\rho)\phi_{targ} + \rho\phi$, where $\rho \in (0,1)$. To mitigate the positive bias in the policy update step, the clipped double Q-learning technique [41] is adopted. The training labels for V networks are modified as $\hat{V}(\boldsymbol{s}_t) = \min_{i=1,2}Q_{\psi_i}(\boldsymbol{s}_t,\boldsymbol{a}_t) - \alpha\ln\pi_\theta(\boldsymbol{a}_t|\boldsymbol{s}_t)$, where two sets of Q networks, $Q_{\psi_1} = \{Q_{\psi_1^l}, Q_{\psi_1^c}\}, Q_{\psi_2} = \{Q_{\psi_2^l}, Q_{\psi_2^c}\}$, are maintained and trained separately. The implementation details of the CSAC algorithm is provided in the Appendix.

E. Policy Gradient for Discrete Action

Discrete control variables are needed to represent the control actions in the VVC problem such as changing the tap/on-off positions of voltage regulators, on-load tap changers, and switchable capacitor banks. The policy gradient of the SAC algorithm designed for a continuous control problem

can not be directly applied for our proposed CSAC. Specifically, in SAC, the sampled actions are reparameterized with $\hat{\boldsymbol{a}}_\theta = \mu_\theta + v_\theta \mathcal{N}(0,1)$, where μ_θ and v_θ are the outputs of mean values and variances from the Gaussian policy network. $\mathcal{N}(0,1)$ is the standard normal distribution. Therefore, the $\hat{\boldsymbol{a}}_\theta$ is differentiable with respect to θ. However, it is no longer true for the discrete actions which are sampled with the output distribution of the policy network. For discrete action space, the policy gradient can be derived in a similar fashion to the policy gradient theorem [28], [44] to maximize the state-value function:

$$\begin{aligned}
&\nabla_\theta V_h^\pi(\boldsymbol{s}) \approx \nabla_\theta \sum_a \pi_\theta(\boldsymbol{a}|\boldsymbol{s})(Q_h^\pi(\boldsymbol{s},\boldsymbol{a}) - \alpha \ln \pi_\theta(\boldsymbol{a}|\boldsymbol{s}))\\
&= \mathop{E}_{a\sim\pi_\theta}[\nabla_\theta \ln \pi_\theta(\boldsymbol{a}|\boldsymbol{s})(Q_h^\pi(\boldsymbol{s},\boldsymbol{a}) - \alpha \ln \pi_\theta(\boldsymbol{a}|\boldsymbol{s}))]\\
&= \mathop{E}_{a\sim\pi_\theta}\Big[\nabla_\theta \ln \pi_\theta(\boldsymbol{a}|\boldsymbol{s})\big(Q_h^\pi(\boldsymbol{s},\boldsymbol{a}) - V_h^\pi(\boldsymbol{s}) - \alpha \ln \pi_\theta(\boldsymbol{a}|\boldsymbol{s})\big)\Big]
\end{aligned} \tag{20}$$

The regularity condition, $\sum_a \pi_\theta(\boldsymbol{a}|\boldsymbol{s})\nabla_\theta \ln \pi_\theta(\boldsymbol{a}|\boldsymbol{s}) = 0$, is used for the derivation of the second line. Note that the loss function for updating the parameters θ of the policy neural network is chosen as (19), whose partial derivative is the negative of (20).

F. Device-Decoupled Policy Network Structure and Ordinal Encoding for Discrete Actions

Since only a single tap position can be chosen by each of the remotely controllable devices for VVC problems, we design the policy neural network with a device-decoupled structure. The input of the policy neural network is the state vector s and the outputs are the probabilities of selecting a tap position for each of the N_c devices. Thus, the dimensionality of the output layer is $\sum_{i=1}^{N_c} |\mathcal{A}_i|$, where $|\mathcal{A}_i|$ denotes the number of tap positions for device i. In this way, the network size only increases linearly with N_c. The j-th action of the i-th device corresponds to the logit output l_{ij} of the last hidden layer of the neural network. The probability p_{ij} of choosing j-th action for the i-th device can be calculated by combining $l_{ij}, 1 \le j \le |\mathcal{A}_i|$ via a softmax function, $p_{ij} = exp(l_{ij})/\sum_j exp(l_{ij})$. The final probability of a tap position combination of all the devices is equal to the product of the probability of each individual device taking its own action, $p(\boldsymbol{a}) = \Pi_{i=1}^{N_c} p_i(a_i)$, where $\boldsymbol{a}$ is the vector of chosen actions across all the devices and a_i is the chosen action of i-th device.

Note that the discrete controls actions of each remotely controllable device can be represented by an ordinal variable. For example, the control actions of an on-load tap changer with 3 tap positions that correspond to turns ratios of 0.95, 1, and 1.05 can be deemed as a discretization of an ordinal variable of turns ratio. Thus, we adopt an ordinal representation [45] for all the discrete actions of a device to encode the natural ordering between the discrete actions.

Specifically, each subset of the logit outputs corresponding to a device is first pre-processed as follows:

$$l'_{ij} = \sum_{m\le j} \ln o_{im} + \sum_{m>j} \ln(1-o_{im}),\ i = 1, 2..., N_c \tag{21}$$

where the sigmoid function is first applied to the logits, $o_{ij} = sigmoid(l_{ij})$, and l'_{ij} is the transformed logit after the ordinal encoding. Then the probability of device i taking action j can be calculated via $p'_{ij} = exp(l'_{ij})/\sum_j exp(l'_{ij})$.

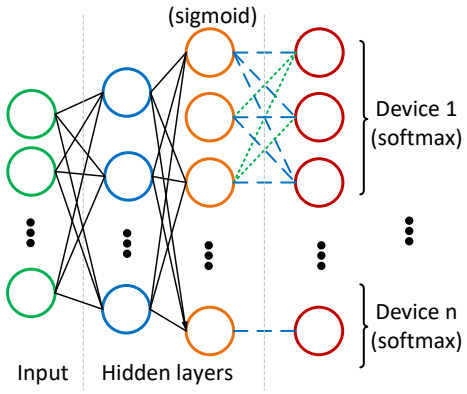


Fig. 1. Device-decoupled structure of the policy neural network

The device-decoupled structure of the policy neural network is depicted in Fig.1, where the long-dashed lines denote the connections associated with $\ln(x)$ and the short-dashed lines denote the connections associated with $\ln(1-x)$. Note that (21) is equivalent to encoding the j-th action of a control device as a vector, $[1, ...1, 0...]$, where the first j elements are set as 1s and the rest of the elements are set as 0s. By introducing an inductive bias which appropriately distinguishes the dissimilarity among the discrete actions, the ordinal encoding further improves the learning efficiency of our proposed CSAC algorithm.

IV. Numerical Study

Numerical studies are carried out on distribution test feeders to validate the sampling efficiency, scalability, optimality, and safety of the proposed CSAC algorithm for solving VVC problems. We also performed a comprehensive comparison between the proposed algorithm and four benchmark algorithms including three RL algorithms and two optimization-based algorithms.

A. Simulation Setup

The IEEE 4-bus, 34-bus and 123-bus distribution test feeders [46] are used in the numerical simulations. In the 4-bus feeder, a voltage regulator is located at node 1 and an on-load tap changer connects node 2 and 3. We add a capacitor bank with 200 kVar rating to node 4. In the 34-bus test feeder, a voltage regulator is at node 800. There are two transformers connecting node 814 to node 850 and node 852 to node 832 respectively. Two capacitors are placed at node 844 (100 kVar) and node 847 (150 kVar). In the 123-bus test feeder, a voltage regulator is at node 150. There are three on-load tap changers, which connect node 10 to node 15, node 160 to node 67, and node 25 to node 26 respectively. Four capacitors are placed at node 83 (200 kVar), node 88 (50 kVar), node 90 (50 kVar), and node 92 (50 kVar). All voltage regulators and on-load tap changers have 11 tap positions, which correspond

to turns ratios ranging from 0.95 to 1.05. The capacitors can be switched on/off remotely and the number of 'tap positions' is treated as 2.

In the initial state, the turns ratios of voltage regulators and on-load tap changers are 1 and the capacitors are switched off. The electricity price C_e is assumed to be $\$40/MWh$. The operating cost per tap change C_j^T is set to be $\$0.1$ for all devices. One year of hourly smart meter energy consumption data [47] from London is used. The aggregated load data is scaled and allocated to each node according to the existing spatial load distribution of the IEEE standard test cases. 10 weeks of randomly selected data are used for out-of-sample testing. The rest of the data are used for training purposes. For DRL approaches, the reward and the cost are derived based on the line losses and nodal voltages calculated from the power flow simulations. For the three IEEE distribution test feeders, when the nodal voltages are within appropriate bounds, the line flow limits are also satisfied. Thus, only the voltage constraints are explicitly stated in the problem formulation. The upper limit for the number of voltage violations $\overline{V^c}$ is set as 0. The parameter settings for the reinforcement learning algorithms are provided in Table I below.

TABLE I
PARAMETER SETTINGS FOR REINFORCEMENT LEARNING ALGORITHMS

Parameters	4-bus	34-bus	123-bus
Size of Hidden Layers	(64, 32)		
Activation Function of Hidden Layers	relu		
Batch Size	256		
Initial Value of λ	0		
Discount Factor γ	0.99		
Temperature Parameter α	0.02	0.02	0.05
Step Size for Q Networks δ_ψ	1e-3		
Step Size for V Networks δ_ϕ	5e-4		
Step Size for π Network δ_θ	1e-3		
Step Size for λ Update δ_λ	1e-5		
Delay Factor ρ	5e-4		

B. Setup of the Benchmark and Our Proposed Algorithms

The deep Q-network (DQN) [32] algorithm, an extension of the tabular Q-learning for the VVC [25], is chosen as the first benchmark RL algorithm. DQN algorithm is one of the most widely used off-policy RL algorithms for solving MDP. In order to apply DQN for CMDP, a penalty term for the voltage violation is added to the reward function as $R_t - C_V R_t^c$, where the penalty coefficient C_V is set as $\$1$ per voltage violation per node. Constrained Policy Optimization (CPO) algorithm, a state-of-the-art RL algorithm for solving CMDP, is chosen as the second benchmark RL algorithm. CPO not only guarantees monotonic policy improvement at each policy iteration step but also ensures constraint satisfaction throughout the training process given that a feasible policy is recovered.

Both our proposed CSAC and the DQN algorithm are off-policy RL algorithms. A single sample is collected at each training step for these two algorithms. On-policy RL algorithms such as CPO typically require a large number of new samples to be collected in order to accurately estimate the state values. In this study, the sampling size of each training step of CPO is set to be 5000, which is determined by gradually increasing the sampling size until the algorithm can achieve a stable and reasonable performance. The length of each episode is set as a week, i.e., 168 hours. The weights of the neural networks are randomly initialized and updated with batch training. The batch size is set as 256.

To illustrate the effectiveness of proposed CSAC method, the SAC algorithm with fixed penalty coefficients is chosen as the third DRL benchmark. Except for removing the update step for λ, the same parameters are chosen as that of the CSAC algorithm. The parameters of neural networks are fine-tuned based on the training performance.

Two benchmark optimization-based algorithms for VVC problems are also implemented. The first benchmark optimization algorithm is implemented based on the single period (one hour) mixed-integer conic programming (MICP), which is the same as the discrete control stage without the chance constraints in [48]. Essentially, a multi-period VVC problem is solved for one hour at a time with the MICP algorithm. The second benchmark optimization algorithm is implemented by extending the single period MICP to multiple periods with model predictive control (MPC) framework as in [12] over a planning horizon of 24 hours. Note that for the optimization-based benchmarks, the actual future load is assumed to be given. The commercial solver GUROBI is used for both benchmark optimization algorithms.

C. Sample Efficiency

Evaluated based on the necessary number of samples to reach a stable solution, the sample efficiency of the proposed CSAC algorithm and the two benchmark RL algorithms is analyzed for the three distribution test feeders in this subsection. The number of training samples collected versus the average weekly return (AVR) on the testing weeks, i.e., the negative of the total operational costs associated with real power losses, tap changes, and voltage violations are shown in the top subfigures of Fig.2-4. The number of weekly voltage violations versus the number of training samples are shown in the bottom subfigures of Fig.2-4. The dark-colored curves are the average performances of 5 random experiments, and the light-colored regions represent the error bounds.

As shown in Fig.2-4, to achieve the same level of performance, our proposed CSAC algorithm needs the least amount of training samples. The on-policy CPO algorithm needs a much higher number of training samples than the off-policy algorithms, CSAC and DQN. In the case of the 4-bus test feeder, CSAC and DNQ only need about 10,000 training samples to achieve stable performance, while CPO requires about 500,000 training samples to achieve stable performance.

The off-policy nature of CSAC algorithm not only significantly improves sample efficiency, but also allows us to reuse historical operational data. In contrast, the on-policy algorithms such as CPO need to generate new samples according to the latest policy at every training step. Moreover, at each step of CPO, a large number of samples need to be collected to form an accurate estimate of the state values.

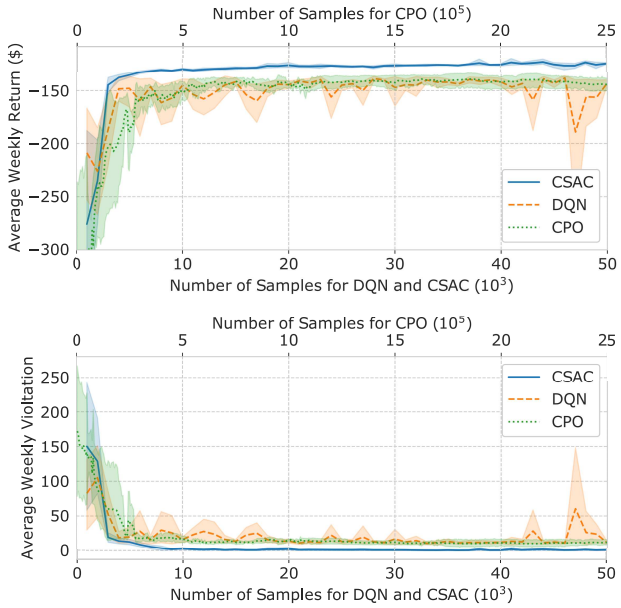


Fig. 2. Average weekly return and voltage violation for 4-bus test feeder

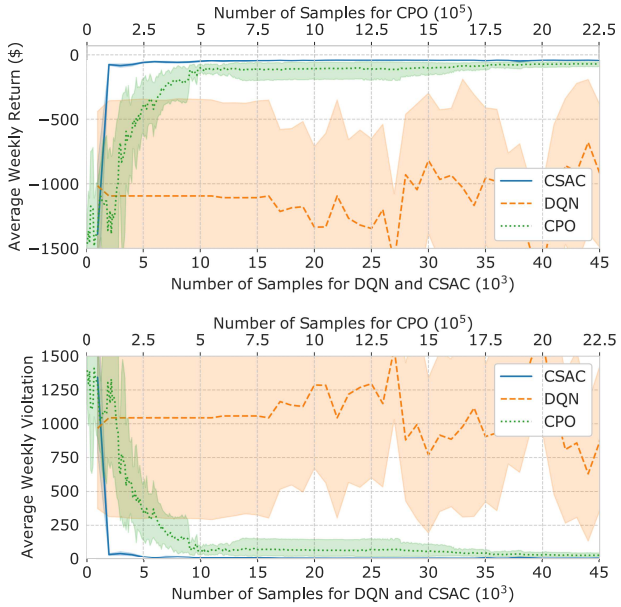


Fig. 3. Average weekly return and voltage violation for 34-bus test feeder

TABLE II
PERFORMANCE COMPARISON OF VOLT-VAR CONTROL ALGORITHMS

Algorithm	AVR ($)			AVV		
	4-bus	34-bus	123-bus	4-bus	34-bus	123-bus
DQN	-140.22	-680.09	N/A†	10.6	630.40	N/A†
CPO	-139.27	-71.78	-68.88	9.6	27.34	1.12
CSAC	-126.58	-42.39	-57.43	0.18	0.06	0
MPC	-122.86	N/A*	N/A*	0	N/A*	N/A*
MICP	-133.26	-44.51	-66.99	0	0	0

* can not find a solution of a rolling step in 4 hours.
† can not finish one epoch of training in 10 hours.

D. Optimality, Constraint Satisfaction, and Scalability

The AVRs and the number of weekly average voltage constraints violations (AVVs) during the testing weeks of the proposed CSAC algorithm, the two benchmark RL algorithms, and the two benchmark optimization algorithms are shown in Table II. The results of the RL algorithms are the averaged performances of 5 experiments, each with a different random seed.

As shown in the table, our proposed CSAC algorithm achieves the highest return among all algorithms for the 34-bus and 123-bus test feeders and the second highest return for the 4-bus test feeder. As the size of the distribution feeder increases, the advantage of our proposed CSAC algorithm becomes more pronounced. Furthermore, our proposed CSAC algorithm can satisfy the voltage constraints almost all the time whereas the other benchmark RL algorithms may lead to significant voltage violations. Minor voltage violations do occur in the 4-bus and 34-bus test feeders when our proposed CSAC algorithm is used. However, the average voltage violation magnitude is much smaller than 0.01 per unit.

Although the MPC extension of the MICP algorithm achieves a better solution on the 4-bus test feeder, it is not scalable and can not find a solution of a single rolling step with 4 hours of computation time for both the 34-bus and the 123-bus systems. Similarly, the DQN algorithm is also not scalable and can not obtain a solution for the 123-bus system within a reasonable amount of time. This is because the number of Q values which need to be calculated for each greedy action selection, $\Pi_i^{N_c}|A_i|$, increases quickly with the number of controllable devices. In our proposed CSAC algorithm, the policy function can be approximated with a neural network whose size increases linearly with the number of devices as presented in III-F. Therefore, our proposed device-decoupled encoding approach has much better scalability. Note that the same device-decoupled network structure was applied on the CPO algorithm, where the trust region constraint is enforced to limit the KL-divergence between the previous policy and the up-

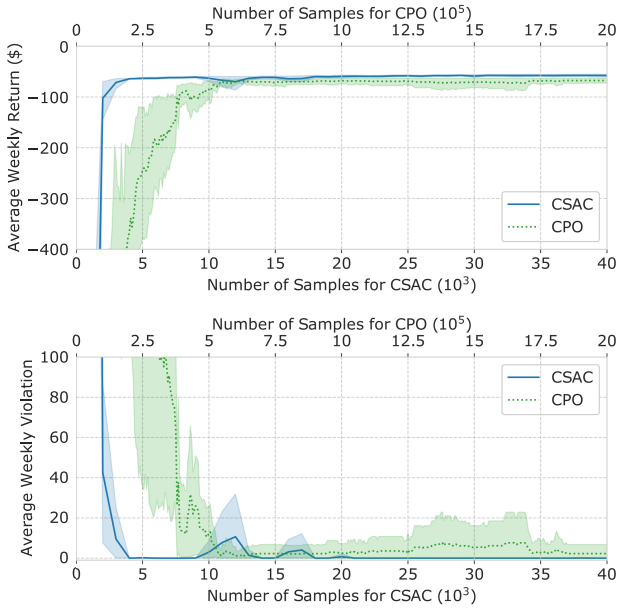


Fig. 4. Average weekly return and voltage constraints violation for 123-bus test feeder

dated policy, $KL(\pi', \pi) = \sum_a \pi'(\boldsymbol{a}|\boldsymbol{s}) \ln(\pi'(\boldsymbol{a}|\boldsymbol{s})/\pi(\boldsymbol{a}|\boldsymbol{s})) \leq \delta$. The total KL-divergence can be decomposed with respect to each device as $KL(\pi', \pi) = \sum_i^{N_c} KL(\pi'_i, \pi_i)$.

TABLE III
COMPARISON BETWEEN CSAC AND SAC FOR VOLT-VAR CONTROL

Algorithm	AVRwV ($)			AVV		
	4-bus	34-bus	123-bus	4-bus	34-bus	123-bus
CSAC	-126.40	-42.33	-57.43	0.18	0.06	0
SAC ($C_V = 0$)	-112.92	-36.50	-53.67	1020	1815.20	340.80
SAC ($C_V = 0.1$)	-125.49	-39.92	-58.19	69.33	10.62	7.94
SAC ($C_V = 1$)	-128.03	-45.31	-59.10	0.27	0.58	0

The comparison between SAC with different penalty coefficients and CSAC is performed to further demonstrate the effectiveness of the proposed method. The comparison results are summarized in III. When the proposed CSAC is compared to the SAC with a constraint violation penalty factor $C_V = 1$, it is very clear that our proposed algorithm not only produces a higher weekly return without the penalty of voltage violations (AVRwV) and a smaller AVV.

V. CONCLUSION

A model-free DRL algorithm is proposed to solve the VVC problem without depending on accurate and complete distribution network topology and parameter information. The VVC problem is formulated as a CMDP and solved by our proposed CSAC algorithm, which is a safe off-policy DRL algorithm. In the algorithm implementation, the policy network is specially designed with a device-decoupled structure and an ordinal encoding scheme. Numerical studies conducted on the 4-bus, 34-bus, and 123-bus distribution test feeders demonstrate that the proposed algorithm achieves better sample-efficiency, scalability and constraint satisfaction than the state-of-the-art reinforcement learning algorithms and the conventional optimization-based algorithms.

APPENDIX
IMPLEMENTATION DETAILS OF THE CSAC ALGORITHM

The CSAC algorithm includes a total of 9 neural networks including $Q_{\psi_1^l}$, $Q_{\psi_2^l}$, V_{ϕ^l} and $V_{\phi^l_{targ}}$ associated with the Lagrange function, $Q_{\psi_1^c}$, $Q_{\psi_2^c}$, V_{ϕ^c} and $V_{\phi^c_{targ}}$ associated with the constraint, and the policy neural network. δ_ψ, δ_ϕ and δ_θ are the corresponding updating step sizes. To update the network parameters, the training targets for the Q and V networks are first obtained according to (22) - (26). Note that double Q-learning is used to mitigate the overestimation of Q values in a maximization problem. The max operation is used in (25) to reduce Q^c. Then, the gradient descent update is performed with Adam optimizer to minimize the MSE error with respect to the training targets. The gradient of the loss function (29) is the negative of (20) with expectation over the sampled batch. The delayed update of target V networks is performed in (30). Finally, the Lagrange multiplier λ is updated following (31)

Algorithm 4 CSAC Algorithm - implementation details

Initialize policy network parameters θ, state-value functions V parameters, ϕ^l, ϕ^c, ϕ^l_{targ}, ϕ^c_{targ}, state-action value functions Q parameters ψ_1^l, ψ_2^l, ψ_1^c, ψ_2^c, and Lagrange multiplier λ
repeat
  for each sample step **do**
    Observe state $\boldsymbol{s}_t$ and take action $\boldsymbol{a}_t \sim \pi(\cdot|\boldsymbol{s}_t)$
    Observe the next state $\boldsymbol{s}_{t+1}$, reward R_t and cost R_t^c
    Store data $D = D \cup (\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, R_t, R_t^c)$
  end for
  for each gradient step **do**
    Sample a batch of transitions, B, randomly

REFERENCES

[1] W. Wang, N. Yu, B. Foggo, J. Davis, and J. Li, "Phase identification in electric power distribution systems by clustering of smart meter data," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2016, pp. 259–265.

[2] W. Wang, N. Yu, and Z. Lu, "Advanced metering infrastructure data driven phase identification in smart grid," *GREEN 2017 Forward*, pp. 16–23, 2017.

[3] B. Foggo and N. Yu, "Improving supervised phase identification through the theory of information losses," *IEEE Transactions on Smart Grid*, 2019.

[4] R. Diao, Z. Wang, D. Shi, Q. Chang, J. Duan, and X. Zhang, "Autonomous voltage control for grid operation using deep reinforcement learning," *arXiv preprint arXiv:1904.10597*, 2019.

[5] Q. Huang, R. Huang, W. Hao, J. Tan, R. Fan, and Z. Huang, "Adaptive power system emergency control using deep reinforcement learning," *IEEE Transactions on Smart Grid*, Aug. 2019.

[6] H. Ahmadi, J. R. Mart, and H. W. Dommel, "A framework for Volt-VAR optimization in distribution systems," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1473–1483, May 2015.

[7] P. Li, H. Ji, C. Wang, J. Zhao, G. Song, F. Ding, and J. Wu, "Coordinated control method of voltage and reactive power for active distribution networks based on soft open point," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 4, pp. 1430–1442, Oct. 2017.

[8] M. H. K. Tushar and C. Assi, "Volt-VAR control through joint optimization of capacitor bank switching, renewable energy, and home appliances," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4077–4086, Sept. 2018.

[9] M. B. Liu, C. A. Canizares, and W. Huang, "Reactive power and voltage control in distribution systems with limited switching operations," *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 889–899, May 2009.

[10] Y. Xu, Z. Y. Dong, R. Zhang, and D. J. Hill, "Multi-timescale coordinated Voltage/VAR control of high renewable-penetrated distribution systems," *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4398–4408, Nov. 2017.

[11] W. Zheng, W. Wu, B. Zhang, and Y. Wang, "Robust reactive power optimization and voltage control method for active distribution networks via dual time-scale coordination," *IET Generation, Transmission Distribution*, vol. 11, no. 6, pp. 1461–1471, May 2017.

[12] Z. Wang, J. Wang, B. Chen, M. M. Begovic, and Y. He, "MPC-based Voltage/VAR optimization for distribution circuits with distributed generators and exponential load models," *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2412–2420, Sept. 2014.

[13] M. Falahi, K. Butler-Purry, and M. Ehsani, "Dynamic reactive power control of islanded microgrids," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3649–3657, Nov. 2013.

[14] K. E. Antoniadou-Plytaria, I. N. Kouveliotis-Lysikatos, P. S. Georgilakis, and N. D. Hatziargyriou, "Distributed and decentralized voltage control of smart distribution networks: Models, methods, and future research," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2999–3008, Nov. 2017.

[15] W. Lin, R. Thomas, and E. Bitar, "Real-time voltage regulation in distribution systems via decentralized PV inverter control," in *HICSS*, Jan. 2018.

10: Compute target labels as

$$\hat{Q}^l(\boldsymbol{s}_t, \boldsymbol{a}_t) = R_t - \lambda R_t^c + \gamma V_{\phi^l_{targ}}(\boldsymbol{s}_{t+1}) \tag{22}$$

$$\hat{Q}^c(\boldsymbol{s}_t, \boldsymbol{a}_t) = R_t^c + \gamma V_{\phi^c_{targ}}(\boldsymbol{s}_{t+1}) \tag{23}$$

$$\hat{V}^l(\boldsymbol{s}_t) = \min_{i=1,2} Q_{\psi_i^l}(\boldsymbol{s}_t, \hat{\boldsymbol{a}}_t) - \alpha \ln \pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t) \tag{24}$$

$$\hat{V}^c(\boldsymbol{s}_t) = \max_{i=1,2} Q_{\psi_i^c}(\boldsymbol{s}_t, \hat{\boldsymbol{a}}_t) - \alpha \ln \pi_\theta(\hat{\boldsymbol{a}}_t|\boldsymbol{s}_t) \tag{25}$$

$$\hat{\boldsymbol{a}}_t \sim \pi_\theta(\cdot|\boldsymbol{s}_t) \tag{26}$$

11: Update ψ_1^l, ψ_2^l, ψ_1^c, ψ_2^c by a gradient descent step $\psi_i^j = \psi_i^j - \delta_{\psi^j} \nabla_\psi f(Q_i^j), \forall i = 1, 2, j = l, c$

$$f(Q_i^j) = \frac{1}{|B|} \sum_B (Q_{\psi_i^j} - \hat{Q}^j)^2 \tag{27}$$

12: Update ϕ^l, ϕ^c by a gradient descent step $\phi^j = \phi^j - \delta_{\phi^j} \nabla_\phi f(V^j), \forall j = l, c$

$$f(V^j) = \frac{1}{|B|} \sum_B (V_{\phi^j} - \hat{V}^j)^2 \tag{28}$$

13: Update θ by a gradient descent step $\theta = \theta - \delta_\theta \nabla_\theta f(\pi)$

$$f(\pi) = \frac{1}{|B|} \sum_B \ln \pi_\theta(\hat{\boldsymbol{a}}|s) \big(\alpha \ln \pi_\theta(\hat{\boldsymbol{a}}|s) - Q_{\psi_1^l}(s, \hat{\boldsymbol{a}}) + V_{\phi^l}(\boldsymbol{s}) \big), \hat{\boldsymbol{a}} \sim \pi_\theta(\cdot|s) \tag{29}$$

14: Update target V function parameters ϕ^l_{targ} and ϕ^c_{targ} with delay factor $\rho^j \in (0, 1)$

$$\phi^j_{targ} = (1 - \rho^j)\phi^j_{targ} + \rho^j \phi^j, j = l, c \tag{30}$$

15: Update target λ with step size δ_λ

$$\lambda = [\lambda + \frac{\delta_\lambda}{|B|} \sum_B (V_{\phi^c} - \overline{V^c})]^+ \tag{31}$$

16: **end for**

17: **until** converge

[16] H. Zhu and H. J. Liu, "Fast local voltage control under limited reactive power: optimality and stability analysis," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3794–3803, Sept. 2016.

[17] H. Zhu and N. Li, "Asynchronous local voltage control in power distribution networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2016, pp. 3461–3465.

[18] H. J. Liu, W. Shi, and H. Zhu, "Distributed voltage control in distribution networks: online and robust implementations," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6106–6117, Nov. 2018.

[19] S. Bolognani, R. Carli, G. Cavraro, and S. Zampieri, "Distributed reactive power feedback control for voltage regulation and loss minimization," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 966–981, Apr. 2015.

[20] G. Cavraro and R. Carli, "Local and distributed voltage control algorithms in distribution networks," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1420–1430, Mar. 2018.

[21] B. Zhang, A. Y. S. Lam, A. D. Domnguez-Garca, and D. Tse, "An optimal and distributed method for voltage regulation in power distribution systems," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1714–1726, Jul. 2015.

[22] A. Abessi, V. Vahidinasab, and M. S. Ghazizadeh, "Centralized support distributed voltage control by using end-users as reactive power support," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 178–188, Jan. 2016.

[23] L. Yu, D. Czarkowski, and F. de Leon, "Optimal distributed voltage regulation for secondary networks with DGs," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 959–967, Jun. 2012.

[24] J. G. Vlachogiannis and N. D. Hatziargyriou, "Reinforcement learning for reactive power control," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1317–1325, Aug. 2004.

[25] Y. Xu, W. Zhang, W. Liu, and F. Ferrese, "Multiagent-based reinforcement learning for optimal reactive power dispatch," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1742–1751, Nov. 2012.

[26] H. Xu, A. D. Domínguez-García, and P. W. Sauer, "Optimal tap setting of voltage regulation transformers using batch reinforcement learning," *arXiv*, Jul. 2018.

[27] Y. Gao, J. Shi, W. Wang, and N. Yu, "Dynamic distribution network reconfiguration using reinforcement learning," in *IEEE SmartGridComm*, Oct. 2019, pp. 1–7.

[28] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[29] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv*, Sept. 2015.

[30] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," in *ICML*, vol. 37, 2015, pp. 1889–1897.

[31] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *ICML*, vol. 80, Stockholmsmssan, Stockholm Sweden, Jul. 2018, pp. 1861–1870.

[32] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[33] H. v. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *AAAI*, Feb. 2016, pp. 2094–2100.

[34] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *ICML*, vol. 70, Aug. 2017, pp. 22–31.

[35] W. Wang, N. Yu, J. Shi, and Y. Gao, "Volt-VAR control in power distribution systems with deep reinforcement learning," in *IEEE SmartGridComm*, Oct. 2019, pp. 1–7.

[36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv*, Jul. 2017.

[37] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *ICML*, vol. 48, Jun. 2016, pp. 1928–1937.

[38] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *ICML*, vol. 70, Aug. 2017, pp. 1352–1361.

[39] B. D. Ziebart, "Modeling purposeful adaptive behavior with the principle of maximum causal entropy," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA, 2010.

[40] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, "Bridging the gap between value and policy based reinforcement learning," in *NIPS*, Feb. 2017, pp. 2775–2785.

[41] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *ICML*, vol. 80, Stockholmsmssan, Stockholm Sweden, Jul. 2018, pp. 1587–1596.

[42] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," *arXiv*, May 2018.

[43] V. Borkar, "An actor-critic algorithm for constrained Markov decision processes," *Systems and Control Letters*, vol. 54, no. 3, pp. 207–213, Mar. 2005.

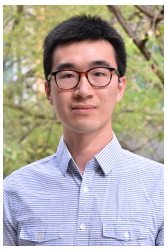
[44] T. Degris, M. White, and R. S. Sutton, "Off-policy actor-critic," *arXiv*, Jun. 2012.

[45] Y. Tang and S. Agrawal, "Discretizing continuous action space for on-policy optimization," *arXiv*, Jan. 2019.

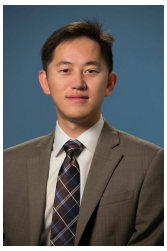
[46] W. H. Kersting, "Radial distribution test feeders," in *IEEE Power Engineering Society Winter Meeting*, vol. 2, Jan. 2001, pp. 908–912.

[47] U. P. Networks. Smart meter energy consumption data in London households. https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households.

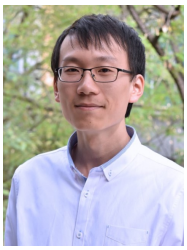
[48] F. U. Nazir, B. C. Pal, and R. A. Jabr, "A two-stage chance constrained Volt/VAR control scheme for active distribution networks with nodal power uncertainties," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 314–325, Jan. 2019.



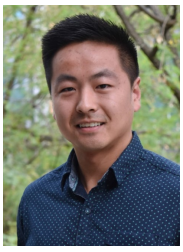
Wei Wang (S'17) received the B.S. degree in Electrical Engineering from the Huazhong University of Science and Technology, Wuhan, China and the M.S.E. degree in Electrical and Computer Engineering from the University of Michigan, Ann Arbor, USA in 2012 and 2014 respectively. He is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at University of California, Riverside, USA. His research interests include optimization and machine learning applications in power systems.



Nanpeng Yu (M'11-SM'16) received his B.S. in Electrical Engineering from Tsinghua University, Beijing, China, in 2006. Dr. Yu also received his M.S. and Ph.D. degree in Electrical Engineering from Iowa State University, Ames, IA, USA in 2007 and 2010 respectively. He is currently an Associate Professor in the department of Electrical and Computer Engineering at University of California, Riverside, CA, USA. His current research interests include machine learning theory, big data analytics in smart grid, electricity market design, and smart energy communities. Dr. Yu is an Editor of IEEE Transactions on Smart Grid, IEEE Transactions on Sustainable Energy, and International Transactions on Electrical Energy Systems.



Yuanqi Gao (S'16) received the B.E. degree in electrical engineering from Donghua University, Shanghai, China in 2015. He is currently working toward the Ph.D. degree in electrical and computer engineering with the University of California, Riverside, CA, USA. His research interests include big data analytics in smart grids.



Jie Shi (S'19) received the B.S. degree in Automation from Shenyang University of Technology, Shenyang, China in 2012. He received the M.S. degree in Control Theory & Engineering from Southeast University, Nanjing, China in 2015. He is currently working toward the Ph.D. degree in Electrical and Computer Engineering with the University of California, Riverside, CA, USA. His research interests include smart city, machine learning, and time series analysis.