Sensitivity-based Heterogeneous Ordered Multi-agent Reinforcement Learning for Distributed Volt-Var Control in Active Distribution Network

Xiaodong Zheng, Shixuan Yu, Hui Cao, Tianzhuo Shi, Shuangsi Xue, Tao Ding

***Abstract*—As power grids expand, maintaining stable voltage and minimizing losses become increasingly crucial. Meanwhile, the widespread use of heterogeneous devices in modern distribution systems necessitates effective multi-device coordination. This issue is exacerbated by the integration of intermittent renewable sources (e.g., solar and wind), which introduce voltage fluctuations. To tackle these challenges, this paper proposes a novel Sensitivity-based Heterogeneous Ordered Multi-agent Reinforcement Learning (SHOM) method for Volt-Var Control (VVC) in Active Distribution Networks (ADNs). By leveraging voltage-reactive sensitivity to explicitly guide sequential policy updates, SHOM ensures a monotonic improvement in control strategies under heterogeneous, networked constraints. Experimental results on IEEE test feeders demonstrate that the proposed approach achieves superior voltage regulation and lower power losses compared to existing methods.**

***Index Terms*—Heterogeneous, Multi-agent Reinforcement Learning, Volt-Var Control, Active Distribution Network.**

I. Introduction

Modern power grids incorporate a variety of energy sources. In recent years, more and more Renewable Energy Sources (RES)[1] have been added to distribution networks. This brings benefits such as better environmental sustainability, lower costs, stronger grid resilience, more energy diversity, and job opportunities. Although there are many advantages to integrating RES, it is vital to note that different power generation types can result in different power quality problems. One example is that the local weather condition can greatly interfere with the solar photovoltaics (PV) and wind power generation systems.

Given the challenges posed by the incorporation of RES, Volt-Var Control (VVC)[2–5] is a pivotal technology in optimizing the Active Distribution Network (ADN). Through optimally dispatching electrical equipment such as voltage regulators, switchable capacitors, and controllable batteries, VVC enables the ADN to dynamically adjust to voltage fluctuations and maximize the energy utilization efficiency of the grid. To optimize the VVC process, voltage-reactive sensitivity (V-Q sensitivity) is a suitable index to quantify how reactive power changes affect node voltages, allowing for more targeted adjustments to control devices, which improves the efficiency of the grid.

Conventionally, the control strategy for these voltage regulation devices is often formulated as an Optimal Power Flow (OPF) problem [6, 7] which can be solved using programming methods such as linear programming, nonlinear programming and mixed integer programming [8]. However, these methods require a fully observable environment, demanding real-time access to both equipment control parameters and complete system states. In practice, this is usually hard to achieve due to massive latent variables such as load fluctuations, temporary topological changes, and limited sensor deployments.

Furthermore, conventional methods for VVC problem in distribution networks heavily rely on communication infrastructures and incur substantial real-time computational overhead. As the network scale increases, the communication and computation burdens for these methods escalate exponentially, resulting high system delay and an over-concentrated computing structure for large-scale systems. This limitation calls for better alternative methods that can effectively address the VVC problem while maintaining a distributed structure which has stronger system robustness.

Inspired by the above insights, researchers have recently developed Deep Reinforcement Learning (DRL) methods [9–11] to tackle VVC problems due to their self-adaptability and high sample efficiency [12–14]. [12] has proposed an Safe Deep Reinforcement Learning(SDRL) algorithm based on the Constrained Policy Optimization (CPO) method [13] to address the optimal operation problem in distribution networks. This study formulates a bi-objective optimization problem which maximize the returns and minimizing voltage constraint violations, considering factors such as substation capacity, voltage, and battery storage limitations. To coordinate the operations of distinct equipment, [14] divides the time horizon into slow and fast timescales. This timescale segmentation facilitates the coordination between capacitor banks (CBs) and inverters for reactive power compensation. Compared to single agent RL methods mentioned above, Multi-agent Reinforcement Learning (MARL)[15–18] is capable of capturing the distributed nature of the power network in a better way since each agent in the group not only optimizes for local objectives but also contributes to the global optimum through joint decision-making. By adopting the Centralized Training Decentralized Execution(CTDE) framework[19], MARL algorithms

This work was supported by the National Natural Science Foundation of China Grant 52277123. *(Corresponding author: Hui Cao.)*

Xiaodong Zheng, Shixuan Yu, Hui Cao, Tianzhuo Shi, Shuangsi Xue, and Tao Ding are with the Shaanxi Key Laboratory of Smart Grid and the State Key Laboratory of Electrical Insulation and Power Equipment, School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: zxd_xjtu@stu.xjtu.edu.cn; rokey001@stu.xjtu.edu.cn; huicao@mail.xjtu.edu.cn; stzdeyx@stu.xjtu.edu.cn; xssxjtu@stu.xjtu.edu.cn; tding15@mail.xjtu.edu.cn)

allow agents executing actions based on local observations rather than collecting the information to a control center for making centralized scheduling, which can prevent potential system level failure caused by some individual agents.

[2] proposes a Multi-agent Constrained Soft Actor-Critic (MACSAC) algorithm to extend the Soft Actor-Critic (SAC) algorithm [20] to solve the VVC problem in the multi-agent environment using the maximum entropy technique. [21] derives an analytical formulation that is applied to each agent using the Deep Deterministic Policy Gradient (DDPG) approach to fit the policy network. [15] proposes an ORT-VVC method using DC-MADRL to optimize voltage stability and reduce power losses in PV-rich distribution networks via multi-agent version of DDPG (MADDPG).

In the above literature, there is a common assumption that the *homogeneity* of agents [19, 22], which implies that all agents are identical. This assumption is often achieved through parameter-sharing techniques, allowing agents to share both action spaces and policy network parameters [23]. However, this method significantly restricts the applicability of the model to real-world scenarios where there are devices with ***heterogeneous*** actions. Solely relying on parameter sharing can lead to suboptimal outcomes[24]. Besides, for VVC problem some algorithms like [3, 14, 21, 25] typically separate the problem the VVC problem into different phases, such as state estimation and control decision-making. This separation can lead to a loss of system integrity and makes it challenging to achieve globally optimal solutions.

Furthermore, the homogeneity assumption presents challenges when addressing VVC problems in online training. In an online training environment, this lack of steady improvement assurance is also hazardous and could lead to critical failures of the system, particularly as agents take stochastic actions due to immature policies in the early exploration stage[26]. This makes it more urgent to develop more robust VVC MARL algorithms that ensures global performance is consistent with local performance.

In this paper, a novel Sensitivity-based Heterogeneous Ordered Multi-agent Reinforcement Learning (SHOM) approach is presented for VVC problem in Active Distribution Networks, where voltage-reactive sensitivity is leveraged to guide the explicit sequential updating of heterogeneous agents under networked constraints. Compared to the existing methods, our contributions are summarized as follows:

1) Contrary to existing MARL algorithms [2, 21, 27] that are limited to adapting to a single type of device at a time, our proposed algorithm is capable of simultaneously controlling heterogeneous device types. This broadens the adaptability for volt-var control in active distributed networks where coordinating different equipment is crucial.
2) To cope with heterogeneous device in the ADN, we designed a training structure based on CTDE framework, as shown in Fig. 3. This structure deeply combines physical device characteristics with sequentially update methods, ensuring the validity of the training. At the same time, we detail how to convert the heterogeneous system state into the reward function, and add the discrete constraints of the device into the reward function as penalty terms to exploit the limit of the algorithm.
3) We designed an updating order selector that associates the updating order of agents with voltage-reactive power sensitivity. This selector helps the model learn time-related characteristics and features. Moreover, since it utilizes active guidance, the algorithm can explicitly evolve and adapt to temporal patterns within the system, thereby enhancing its ability to predict and respond to dynamic changes.

The remainder of this article is organized as follows: Section II introduces the VVC model and outlines the general objectives of the power system. In Section III, we provide an overview of multi-agent reinforcement learning and the definition of Markov games. Section IV presents our proposed VVC algorithm, SHOM, along with its derivations. Experimental results, including performance evaluation and deployment comparisons with other algorithms, are detailed in Section V. Finally, the conclusions are summarized in Section VI.

II. Problem Formulation and Preliminaries

In this section, we begin by introducing the system model based on the branch-flow formulation. Next, we derive the optimization objective and discuss the CTDE framework in detail.

A. Volt-Var Control Model

Consider a radial structure distribution grid represented by

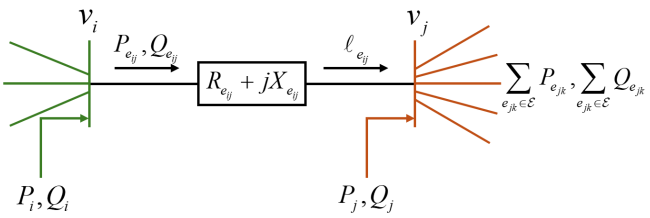


FIG. 1: The Branch Flow model

graph $G = (\mathcal{N}, \mathcal{E})$ with a set of nodes $\mathcal{N}$ and edges $\mathcal{E}$ as shown in Fig. 1.Each edge $e_{ij} \in \mathcal{E}$ connects nodes i and j, where $i, j \in \mathcal{N}$. For such a power system, let v_i represent the voltage magnitude at node i, $\ell_{e_{ij}}$ denote the squared current magnitude on edge e_{ij}, and $P_{e_{ij}}$ and $Q_{e_{ij}}$ represent the active and reactive power injections to edge e_{ij}, respectively. The well known *Branch Flow Model* [6] is denoted as:

$$P_j = \sum_{e_{jk}\in\mathcal{E}} P_{e_{jk}} - P_{e_{ij}} + R_{e_{ij}}\ell_{e_{ij}} \tag{1a}$$

$$Q_j = \sum_{e_{jk}\in\mathcal{E}} Q_{e_{jk}} - Q_{ij} + X_{e_{ij}}\ell_{e_{ij}} \tag{1b}$$

$$v_j^2 = v_i^2 - 2\left(R_{e_{ij}}P_{e_{ij}} + X_{e_{ij}}Q_{e_{ij}}\right) + \left(R_{e_{ij}}^2 + X_{e_{ij}}^2\right)\ell_{e_{ij}} \tag{1c}$$

$$\ell_{e_{ij}} = \left(P_{e_{ij}}^2 + Q_{e_{ij}}^2\right)/v_i^2 \tag{1d}$$

here, $R_{e_{ij}}$ and $X_{e_{ij}}$ are the resistance and reactance of line e_{ij}, P_j and Q_j represent the total active and reactive power injections at node j (1a-1b), and $\ell_{e_{ij}}$ reflects the current

constraint based on the power flow and voltage at the sending node i as shown in (1d).

For Volt-Var control within a given ADN, let x_i denote the different control variables that affect the system status, and let $f_m(x_i)$ represent the contribution of the control variable x_i to the m-th performance factor of the ADN. Then, the overall system health status $\mathcal{J}$ can be expressed as the accumulated impact of all control variables across M performance factors:

$$\mathcal{J} = \sum_{m\in\mathcal{M}} \sum_{i=1}^{N} f_m(x_i) \tag{2}$$

where $M = |\mathcal{M}|$ is the number of factors that affect the overall performance of the ADN, and N is the total node number.

In this article, we focus on three key factors to encapsulate the performance of the system, i.e., $|\mathcal{M}| = 3$: the count of voltage violations vv, the magnitude of control error ce, and the power loss pl of the grid. To optimize the voltage profile and reduce network power losses while satisfying the physical constraints of grid devices, the Eq. (2) can be rewritten as Eq. (3):

$$\mathcal{J} = \sum_{i}^{N} \left(f_{vv}(x_i) + f_{ce}(x_i) + f_{pl}(x_i)\right) \tag{3}$$

where $f(\cdot)$ is a function that transforms a physical state into a measurable quantity. Specifically:

$$\begin{aligned} f_{vv}(x_i) &= \sum_{t=1}^{T} \left(\sigma_0(v_{i,t} - \mathcal{V}_{\max}) + \sigma_0(\mathcal{V}_{\min} - v_{i,t})\right) \\ f_{ce}(x_i) &= \sum_{t=1}^{T} \left(|\Delta x_{i,t}^{bat}| + |\Delta x_{i,t}^{reg}| + |\Delta x_{i,t}^{cap}|\right) \\ f_{pl}(x_i) &= \sum_{e_{ij}\in\mathcal{E}} \left(R_{e_{ij}} \ell_{e_{ij}}\right) \end{aligned} \tag{4}$$

where $\sigma_0(\cdot)$ is the linear rectification function represents $\max(0,\cdot)$, vv, ce, pl represent voltage violation, control impact, and power-loss, respectively. The terms $\Delta x_{i,t}^{bat}$, $\Delta x_{i,t}^{reg}$, and $\Delta x_{i,t}^{cap}$ respectively represent the changes in variables between two successive time step t and $t+1$. The optimization problem (3) is approached by leveraging three main types of controllable heterogeneous devices: regulators with switchable taps $\mathcal{R}_i \in \mathcal{N}_{reg}$, capacitor banks $\mathcal{C}_i \in \mathcal{N}_{cap}$, and batteries $\mathcal{B}_i \in \mathcal{N}_{bat}$. These devices coordinately contribute to the balance of the network and constitute the agent set $\mathcal{N}_a$, i.e. $N_a = |\mathcal{N}_a| = |\mathcal{N}_{reg}| + |\mathcal{N}_{cap}| + |\mathcal{N}_{bat}|$.

Furthermore, we consider two optimizable constraints: the limits on device operable range and the count of device actions. The following constrained optimization problem is formulated:

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \mathcal{J} = \sum_{i=1}^{N} \Big(f_{vv}(x_i) + f_{ce}(x_i) + f_{pl}(x_i)\Big) \tag{5}$$

subject to:

1) Voltage and battery discharge limits:

$$\begin{cases} \mathcal{V}_{\min} \le x_i^{v} \le \mathcal{V}_{\max} \\ \mathcal{P}_{\min}^{\text{dis}} \le x_i^{\text{bat}} \le \mathcal{P}_{\max}^{\text{dis}}, \quad \text{if } i \in \mathcal{N}_{\text{bat}} \end{cases} \tag{6}$$

2) Regulator and capacitor action limits:

$$\begin{cases} \sum_{t=1}^{T} \left|x_i^{\text{reg}}(t+1) - x_i^{\text{reg}}(t)\right| \le \text{tap}_{\max}^{\text{reg}}, \quad \text{if } i \in \mathcal{N}_{\text{reg}} \\ \sum_{t=1}^{T} \left|x_i^{\text{cap}}(t+1) - x_i^{\text{cap}}(t)\right| \le \text{stat}_{\max}^{\text{cap}}, \quad \text{if } i \in \mathcal{N}_{\text{cap}} \end{cases} \tag{7}$$

where $\mathcal{P}_{Max}^{dis}$ and $\mathcal{P}_{min}^{dis}$ are the maximum and minimum discharge power limits of batteries when they are running in the healthy operation interval. $\text{tap}_{\max}^{reg}$ and $\text{stat}_{\max}^{cap}$ are the maximum tap and switch operation numbers for regulators and capacitors.

To optimize (5) under discrete constraints, conventional methods such as MILP, MINLP, and other heuristic algorithms can be used. However, these methods often lack adaptability and are computationally expensive when the system state space getting large. In contrast, MARL-based methods adapt well to dynamic environments.

In this paper, we propose a MARL based method using multiple agents to interact with the environment, i.e. the distribution networks. The ultimate goal for the agents is to jointly achieve the optimization of (5) by optimizing the operation of the network devices.

III. Markov games and System Transformations

This section first introduces the Markov Games framework as the basis for subsequent analysis. Then, the system state over VVC problem is transformed in to MARL framework. Finally, the overall reward function is derived.

A. Markov Games

A Markov Game (MG) defined by the tuple $MG_e = (\mathcal{S}, \boldsymbol{\mathcal{A}}, \mathcal{T}, r, \gamma, \mathcal{N}_a, T)$. Here, $\mathcal{S}$ represents a finite state space shared by all agents, and $\boldsymbol{\mathcal{A}}$ denotes the joint action space, which is the Cartesian Product of individual action spaces $\mathcal{A}_i$, i.e., $\boldsymbol{\mathcal{A}} = \prod_{i=1}^{n} \mathcal{A}_i$. The transition probability function $\mathcal{T} : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to [0, 1]$ defines the probability of transitioning from one state to another given a joint action. The reward function $r : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \mathbb{R}$ assigns rewards to agents based on the current state and joint action. The discount factor γ ranges from 0 to 1, and $\mathcal{N}_a \subseteq \mathcal{N}$ indicates the set of agents. Lastly, T denotes the episode length.

In MGs, each agents follows an action selection pattern called policy π, which maps the current system state to a probability distribution over actions. The goal of the MARL algorithm is to maximize the objective function $\mathcal{J}(\boldsymbol{\pi})$ to obtain the optimal group policy $\boldsymbol{\pi}^*$:

$$\boldsymbol{\pi}^*(\boldsymbol{a}_t|s_t) = \arg\max_{\boldsymbol{\pi}} \mathcal{J}(\boldsymbol{\pi}) \tag{8}$$

where the objective function is denoted as the expectation of accumulated rewards:

$$\mathcal{J}(\boldsymbol{\pi}) := \mathbb{E}_{\mathbf{s}_{0:\infty}\sim\rho_{\pi}^{0:\infty}, \boldsymbol{a}_{0:\infty}\sim\boldsymbol{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{r}_t\right] \tag{9}$$

here, $\boldsymbol{\rho_\pi}$ is the marginal state distribution defined as $\sum_{t=0}^{\infty} \gamma^t \rho_{\boldsymbol{\pi}}^t$, and $\boldsymbol{\pi}$ denotes the group policy.

B. System State Transformation

To apply the MG framework to the VVC problem in distribution network, we associate each abstract agent with a

specific controllable device in the ADN. The following parts define the associated spaces between agents and the system devices.

1) *State space*: The system state comprises the bus voltage s_v^i, the regulator tap status $s_{tap}^{\mathcal{R}}$, the capacitor status $s_{stat}^{\mathcal{C}}$, the state of charge(SoC) of batteries $s_{soc}^{\mathcal{B}}$, and the battery discharge power $s_{dp}^{\mathcal{B}}$. Each of these states is represented according to its device type, either as a binary vector (e.g., $s_{tap}^{\mathcal{R}}$ and $s_{stat}^{\mathcal{C}}$) or as a continuous value normalized to a bounded range (e.g., s_v^i, $s_{soc}^{\mathcal{B}}$, and $s_{dp}^{\mathcal{B}}$). For instance, the bus voltage is represented using the per-unit value based on the base voltage. Thus, the state space is formulated as Eq. (10) :

$$\boldsymbol{s} = (\boldsymbol{s_v^i}, \boldsymbol{s^{i_f}}) \tag{10a}$$

$$\boldsymbol{s_v^i} = \textit{Concats}\left(s_v^i\right), \forall i \in \mathcal{N} \tag{10b}$$

$$\boldsymbol{s^{i_f}} = \textit{Concats}\left(s_{tap}^{\mathcal{R}_i}, s_{stat}^{\mathcal{C}_i}, s_{soc}^{\mathcal{B}_i}, s_{dp}^{\mathcal{B}_i}\right) \tag{10c}$$

where $\mathcal{R}_i \in \mathcal{N}_{reg}, \forall \mathcal{B}_i \in \mathcal{N}_{bat}, \forall \mathcal{C}_i \in \mathcal{N}_{cap}$. Here, we simplify the time step subscript t since all states depend on time. The state $\boldsymbol{s}$ is composed of two parts. $\boldsymbol{s_v^i}$ represents the bus voltage for all nodes, while $\boldsymbol{s^{i_f}}$ only represents the device state for operational nodes, which in this context are the agents.

2) *Action Space:* The action space for agents is defined by limited control variables that can affect system status. Each capacitor operates with a binary on/off action $a^{\mathcal{C}} \in \{0, 1\}$. Regulators have N_{tap} taps, each controllable as on or off, forming an N_{tap}-dimensional action vector $a^{\mathcal{R}}$. Similar to the regulator, discrete batteries use a binary vector $a_d^{\mathcal{B}}$ with N_{bat} dimensions, matching their N_{bat} discharging levels.

3) *State Transition Function:* The state transition is represented by a transition function:

$$\mathcal{T}(s_{t+1} \mid s_t, \boldsymbol{a_t}) = \begin{cases} f_{\text{load-flow}}(s_{v,t}^i, a_t^{\mathcal{R}}, a_t^{\mathcal{C}}, \text{load}_t), \\ a_t^{\mathcal{R}}, \quad \textit{regulator} \\ a_t^{\mathcal{C}}, \quad \textit{capacitor} \\ f_{soc}^{bat}(s_{soc,t}^{\mathcal{B}}, a_t^{\mathcal{B}}), \quad \textit{soc} \\ f_{dp}^{bat}(s_{dp,t}^{\mathcal{B}}, a_t^{\mathcal{B}}), \quad \textit{discharge power} \end{cases} \tag{11}$$

where $f_{\text{load-flow}}(s_{v,t}^i, a_t^{\mathcal{R}}, a_t^{\mathcal{C}}, \text{load}_t)$ is the power flow equation follows Eq. (1) and could be solved to get the new node voltage $s_{v,t+1}^i$. $a_t^{\mathcal{R}}$ and $a_t^{\mathcal{C}}$ represent action vectors of regulators and capacitors. f_{soc}^{bat} and f_{dp}^{bat} are the transition functions for SOC and discharge power, which can be defined as:

$$f_{soc}^{bat}(s_{soc,t}^{\mathcal{B}}, a_t^{\mathcal{B}}) = a_t^{\mathcal{B}} * 3600/\text{soc}_{\max} + s_{soc,t}^{\mathcal{B}} \tag{12a}$$

$$f_{dp}^{bat}(s_{dp,t}^{\mathcal{B}}, a_t^{\mathcal{B}}) = a_t^{\mathcal{B}} \tag{12b}$$

note that $\text{soc}_{\max}$ in Eq. (12a) represents the upper limit of the battery soc.

C. Objectives Derivation

As described by the Eq. (5) , the primary objective of volt-var control in an ADN is to minimize the combined effects of voltage violations, control errors, and power losses under device constraint. In this subsection, we derive the individual penalty components and present the overall reward function.

1) Voltage Violation Penalty:

We define the penalty for node voltages that exceed $\pm 5\%$ of the nominal per-unit voltage thresholds. The cost function for voltage violations is defined as:

$$\sum_{i \in \mathcal{N}} f_{vv}^t(s_t^i, \boldsymbol{a}_t) = \gamma_{vv} \sum_{i=1}^{N} \left(\max\left(\Delta\mathcal{V}_{\max}, 0\right) + \max\left(\Delta\mathcal{V}_{\min}, 0\right)\right) \tag{13}$$

where γ_{vv} is a scaling coefficient. The $\Delta\mathcal{V}_{\max}$ and $\Delta\mathcal{V}_{\min}$ measure the extent that node i violate the upper and lower voltage limits, respectively:

$$\begin{aligned} \Delta\mathcal{V}_{min} &:= \mathcal{V}_{min} - \min_{p \in \mathbf{P(i)}} v_{i,p}(s_v^i(t), \boldsymbol{a}(t)) \\ \Delta\mathcal{V}_{\max} &:= \max_{p \in \mathbf{P(i)}} v_{i,p}(s_v^i(t), \boldsymbol{a}(t)) - \mathcal{V}_{max} \end{aligned} \tag{14}$$

here, p refers to the phases $\mathbf{P}(i)$ of node i.

2) Power Loss Penalty:

The power loss penalty is incorporated into the reward as:

$$\sum_{i \in \mathcal{N}} f_{pl}^t(s_t^i, \boldsymbol{a}_t) = \beta_{pl} \frac{PL_t}{TP_t} \tag{15}$$

where PL_t is the power loss at time t of the ADN, and TP_t is the total power injected into the grid, and βpl is a scaling coefficient.

3) Excessive Control Penalty:

To reduce the device wear, we penalize the excessive control actions by:

$$\begin{aligned} &\sum_{i_f \in \mathcal{N}_a} f_{ce}^t(s_t^{i_f}, a_t^{i_f}) = \sum_{i_f \in \mathcal{N}_a} \boldsymbol{\eta} \left| s_t^{i_f} - s_{t+1}^{i_f} \right| \\ &= \sum_{\mathcal{C}_i \in \mathcal{N}_{cap}} \eta_{\mathcal{C}} \left| s_t^{\mathcal{C}_i} - s_{t+1}^{\mathcal{C}_i} \right| + \sum_{\mathcal{R}_i \in \mathcal{N}_{reg}} \eta_{\mathcal{R}} \left| s_t^{\mathcal{R}_i^{tap}} - s_{t+1}^{\mathcal{R}_i^{tap}} \right| \\ &+ \sum_{\mathcal{B}_i \in \mathcal{N}_{bat}} \eta_{\mathcal{B}}^{soc} \left| \mathop{\mathbb{I}}_{t=T} \left(s_{t+1}^{\mathcal{B}_i^{soc}} - s_0^{\mathcal{B}_i^{soc}} \right) \right| + \eta_{\mathcal{B}}^{dp} \left| \frac{s_{t+1}^{\mathcal{B}_i^{dp}}}{dp_{max}^{\mathcal{B}_i}} \right| \end{aligned} \tag{16}$$

where $s_{t+1}^{i_f} = T\left(s_t^{i_f}, a_t^{i_f}\right)$ represent the state transition given current state $s_t^{i_f}$ and device action $a_t^{i_f}$. The weights $\eta_{\mathcal{C}}, \eta_{\mathcal{R}}, \eta_{\mathcal{B}}^{soc}, \eta_{\mathcal{B}}^{dp}$ balance the control penalties among different device types. The indicator function $\mathop{\mathbb{I}}_{t=T}$ encourages batteries to restore their state-of-charge(SOC) by the end of each episode.

4) Device Constraint Penalties:

In addition to the direct penalties in the reward function, we also incorporate constraints via Lagrangian multipliers.

$$\begin{aligned} &\sum_{i_f \in \mathcal{N}_a} f_{dcp}(\boldsymbol{s_t^{i_f}}) = \lambda_i^{vv} \sum_{i=1}^{N} \sum_{k=1}^{t} \mathbb{I}(\mathcal{V}_{\min} - s_{v,k}^i) + \mathbb{I}(s_{v,k}^i - \mathcal{V}_{\max}) \\ &+ \lambda_{\mathcal{B}_i}^{bat} \sum_{\mathcal{B}_i \in \mathcal{N}_{\text{bat}}} \sum_{k=1}^{t} \left(\mathbb{I}(\mathcal{P}_{\min}^{dis} - s_{dp,k}^{\mathcal{B}_i}) + \mathbb{I}(s_{dp,k}^{\mathcal{B}_i} - \mathcal{P}_{\max}^{dis}) \right) \\ &+ \lambda_{\mathcal{R}_i}^{reg} \sum_{\mathcal{R}_i \in \mathcal{N}_{\text{reg}}} \mathop{\sigma}_{ReLU} \left(\sum_{k=1}^{t} \left| s_{tap,k+1}^{\mathcal{R}_i} - s_{tap,k}^{\mathcal{R}_i} \right| - \text{tap}_{\max}^{reg} \right) \\ &+ \lambda_{\mathcal{C}_i}^{cap} \sum_{\mathcal{C}_i \in \mathcal{N}_{\text{cap}}} \mathop{\sigma}_{ReLU} \left(\sum_{k=1}^{t} \left| s_{stat,k+1}^{\mathcal{C}_i} - s_{stat,k}^{\mathcal{C}_i} \right| - \text{stat}_{\max}^{cap} \right) \Bigg) \end{aligned} \tag{17}$$

where $\mathop{\sigma}_{ReLU}$ is the ReLU function to ensure the penalty is non-negative and differentiable, and $\lambda_{\mathcal{B}_i/\mathcal{R}_i/\mathcal{C}_i}^{\text{vv/bat/reg/cap}}$s are corresponding coefficients for training the agents. The term $f_{dcp}(\boldsymbol{s_t^{i_f}})$

represents an additional penalty imposed to enforce device constraints.

5) Overall Reward Function

Combining all the penalty components—voltage violation, power losses, excessive control actions, and device constraint penalties—into a single reward function yields:

$$r(\boldsymbol{s_t}, \boldsymbol{a_t}) = -\left(\sum_{i\in\mathcal{N}} \left(f_{vv}^t(\boldsymbol{s_t^i}, \boldsymbol{a}_t) + f_{pl}^t(\boldsymbol{s_t^i}, \boldsymbol{a}_t)\right) + \sum_{i_f\in\mathcal{N}_a} f_{ce}^t(\boldsymbol{s_t^{i_f}}, \boldsymbol{a_t^{i_f}}) + \sum_{i_f\in\mathcal{N}_a} f_{dcp}(\boldsymbol{s_t^{i_f}})\right) \tag{18}$$

where i indexes all nodes in the network and i_f indexes the controllable nodes, hereafter called **agents**.(18) forms the foundation of the general optimization function in (9).

IV. Methods

In this section, we first derive the heterogeneous multi-agent reinforcement monotonic learning in detail. Then, we introduce the order selector, which is a volt-var sensitivity based ranker for the agent update process. Finally, we present the pseudo-code of the SHOM algorithm.

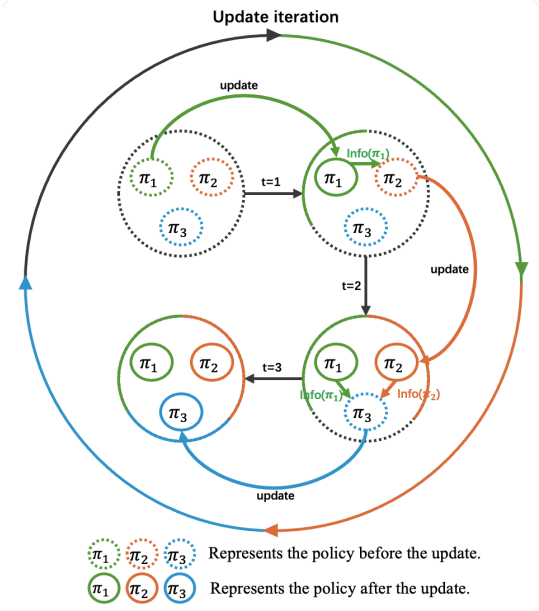


FIG. 2: The sequential updating process, $\pi_{1,2,3}$ is the individual policy, in each iteration, the agent that updates first passes the update information to the agent that updates later

A. A Monotonic Way of Heterogeneous Multi-agent Reinforcement Learning

Let $\mathcal{L}_\pi$ be a surrogate function, where π is the current agent policy and $\hat{\pi}$ the updated policy. According to [24], the surrogate function $\mathcal{L}_\pi(\hat{\pi})$ could serve as a lower bound on the performance index $\mathcal{J}(\hat{\pi})$:

$$\mathcal{J}(\hat{\pi}) \geq \mathcal{L}_\pi(\hat{\pi}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\mathcal{K}_{\max}(\pi, \hat{\pi}), \tag{19}$$

where $\mathcal{K}$ is the KL-divergence between π and $\hat{\pi}$, and other are hyper-parameters. By updating the policy using (20):

$$\pi_{k+1} = \arg\max(\mathcal{L}_\pi(\hat{\pi}) - \zeta\mathcal{K}(\pi_k, \hat{\pi})), \tag{20}$$

and iterating this update by (21), we can get the incremental performance gains:

$$\begin{aligned}\phi_{k+1} = &\arg\max \mathcal{L}_{\pi_{\phi_k}}(\pi) \\ &\text{subject to } \mathbb{E}[\mathcal{K}(\pi_{\phi_{k+1}}, \pi_{\phi_k})] \leq \delta\end{aligned} \tag{21}$$

where ϕ is the actor network parameter, and subscript k represent k-th update iteration.

Now, consider a multi-agent system partitioned into squads indexed by $\widehat{m}$, each containing m sequentially indexed agents $i_{\widehat{m}}$. For convenience, define $\widehat{j'} := \widehat{j-1}$. From [28], the multi-agent advantage can be decomposed as:

$$A_{\boldsymbol{\pi}}^{i_{\widehat{m}}}(s, \boldsymbol{a}^{i_{\widehat{m}}}) = \sum_{j=1}^{m} A_{\boldsymbol{\pi}}^{i_j}(s, \boldsymbol{a}^{i_{\widehat{j'}}}, a^{i_j}), \tag{22}$$

and for disjoint squads $\widehat{l}$ and $\widehat{m}$ in the group:

$$A_{\boldsymbol{\pi}}^{i_{\widehat{m}}}\left(s, \boldsymbol{a}^{j_{\widehat{l}}}, \boldsymbol{a}^{i_{\widehat{m}}}\right) := Q_{\boldsymbol{\pi}}^{j_{\widehat{l}}, i_{\widehat{m}}}\left(s, \boldsymbol{a}^{j_{\widehat{l}}}, \boldsymbol{a}^{i_{\widehat{m}}}\right) - Q_{\boldsymbol{\pi}}^{j_{\widehat{l}}}\left(s, \boldsymbol{a}^{j_{\widehat{l}}}\right) \tag{23}$$

where $Q_{\boldsymbol{\pi}}$ is the joint *Q function*[28] of policy $\boldsymbol{\pi}$.

In Eq. (22), the joint advantage function is decomposed into the sum of individual agents' local advantage functions. In this way, the evaluation of the contribution of each agent to global performance becomes quantifiable, resulting a surrogate function able to update the policy of m-th agent given the previously updated $(m-1)$ agents. Also, to reduce complexity, the PPO[29] clipping method is used, yielding the final surrogate function as:

$$\begin{aligned}\mathcal{L}^{i_m}(\phi) = &\mathbb{E}\left[\min\left(d_k(\phi)M^{i_{\widehat{m}}}(s, \boldsymbol{a}),\right.\right. \\ &\left.\left.\text{clip}\left(d_k(\phi), 1\pm\epsilon\right)M^{i_{\widehat{m}}}(s, \boldsymbol{a})\right)\right]\end{aligned} \tag{24}$$

where $d_k(\phi) = \frac{\pi_{\phi^{im}}^{i_m}(a^i|s)}{\pi_{\phi_k^{im}}^{i_m}(a^i|s)}$ and $M^{i_{\widehat{m'}}} = \frac{\overline{\boldsymbol{\pi}}^{i_{\widehat{m'}}}(\boldsymbol{a}^{i_{\widehat{m'}}}|s)}{\boldsymbol{\pi}^{i_{\widehat{m'}}}(\boldsymbol{a}^{i_{\widehat{m'}}}|s)}\hat{A}(s, \boldsymbol{a})$.

The parameter ϵ adjusts the clipping range and $\hat{A}(s, \boldsymbol{a})$ is an GAE [30] estimator of the joint advantage function. Fig. 2 shows the sequentially updating process for heterogeneous agents, and the optimal policy under monotonic improvements is then obtained by:

$$(\pi^{i_m})^* = \arg\max_{\boldsymbol{\pi_\phi}} \mathcal{L}_{\boldsymbol{\pi}}^{i_m}\left(\overline{\boldsymbol{\pi}}^{i_{\widehat{m'}}}, \hat{\pi}^{i_m}\right). \tag{25}$$

Finally, iteratively incorporating (24) into Eq. (19) leads to the the optimized of $\mathcal{J}$:

$$\phi_{k+1} = \arg\max_{\phi_k} \mathcal{L}(\pi_\phi), \tag{26}$$

Based on the monotonic policy updating scheme described above, we propose a centralized training and decentralized execution(CTDE) control framework tailored for mixed types of devices in volt-var control. In Fig. 3, each agent interacts with the environment to receive partial observations that to update its local policy, while a centralized critic is trained using combined experiences to ensure the global coordination. During execution, agents independently make their action decisions based on local observations and their own policy networks, forming a joint action that makes the system robust against failures in centralized control or computation.

B. Sensitivity-based Updating Order Selector

For Eq. (25), one question is how to determine the order in which agents should update their policy. This question is important since different updating orders lead to different system dynamics. Suitable updating orders can lead to better

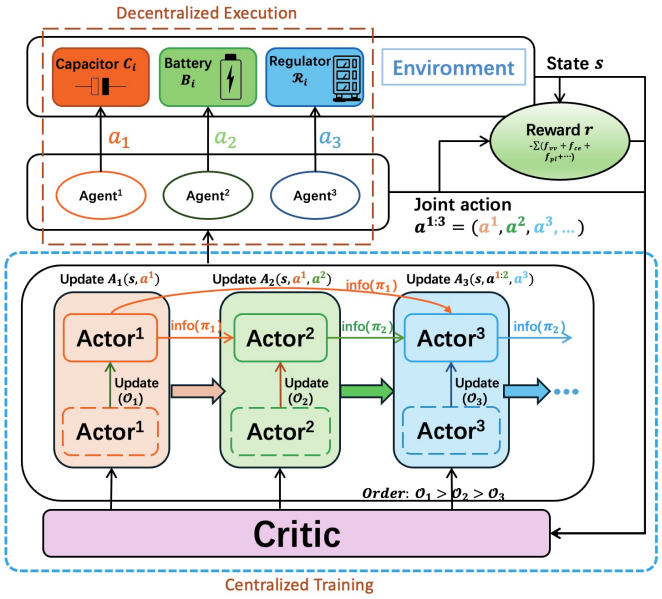


FIG. 3: CTDE framework with sequentially updating of the agents

coordination and more accurately reflect the dynamic changes in system loads. By giving different orders during the update process, the actor networks can learn the ordered joint actions that correspond to changes in the dynamics of the environment.

Building on this idea, we designed an order selector to determine the agent update sequence during the training stage. This selector aims to determine the agent that needs to be updated first based on the voltage-reactive power (V-Q) sensitivity of nodes. The voltage-reactive power sensitivity matrix $\mathbf{S}_{VQ}$ can be derived in Eq. (27):

$$\begin{cases} \begin{bmatrix} Q \\ P \end{bmatrix} = \begin{bmatrix} J_{Q\theta} & J_{QV} \\ J_{P\theta} & J_{PV} \end{bmatrix} \cdot \begin{bmatrix} V \\ \theta \end{bmatrix} \\ \mathbf{S}_{VQ} = \frac{\partial V}{\partial Q} = \left(\frac{\partial Q}{\partial V} - \frac{\partial Q}{\partial \theta} \left(\frac{\partial P}{\partial \theta} \right)^{-1} \frac{\partial P}{\partial V} \right)^{-1} \end{cases} \tag{27}$$

The V-Q sensitivity could be used to evaluate the voltage stability of nodes. A higher accumulated $\mathbf{S}_{VQ}$ value at a node indicates a stronger influence of the voltage change on reactive power, suggesting that the node is more sensitive to voltage stability. Therefore, during the agent updating, agents with higher $\mathbf{S}_{VQ}$ values should be given higher priority. The V-Q sensitivity for each node during the episode is calculated:

$$S_{VQ,t}(i) = \sum_{j=1}^{N} |\mathbf{S}_{VQ,t}(i,j)| \tag{28}$$

$S_{VQ}(i)$ is a scalar that represents the V-Q sensitivity of node i relative to other nodes at the end of the episode.

Given the time-dependent nature of the load data and environmental dynamics, we incorporate a weight w_{QV} into the node V-Q sensitivity to capture the importance of each time step within an episode. The weight is calculated as the ratio of each time-step reward, r_t, to the total episode reward, r_T, which ensures the model to adapt dynamically to temporal load variations and improve predictive accuracy and responsiveness.

$$w_{VQ,t} = \frac{r_t}{r_T} = \frac{\boldsymbol{r}(\boldsymbol{s_t}, \boldsymbol{a_t})}{\sum_{t=1}^{T} \boldsymbol{r}(\boldsymbol{s_t}, \boldsymbol{a_t})} \tag{29}$$

Hence, the updating order is finally determined by masking the pure-load node and sorting the agent node by its weighted V-Q sensitivity.

$$\mathcal{O}(i) = \underset{n \in \mathcal{N} \setminus \mathcal{N}_a}{\text{Mask}} \left(\text{Sort} \left(\sum_{t}^{T} w_{VQ,t} \cdot S_{VQ,t}(i) \right) \right) \tag{30}$$

where $\mathcal{O}(i)$ is the sorted order of agent i in an episode, and this order is going to be used in the next update process. The operation $\underset{n \in \mathcal{N} \setminus \mathcal{N}_a}{\text{Mask}}$ exclusively masks non-agent nodes as they are not controllable.Fig. 4 shows the updating changing process of order selector.

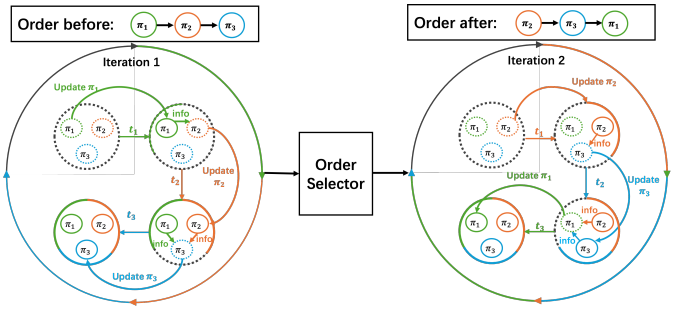


FIG. 4: The order selector rearranges the policy updating order

Algorithm 1 Sensitivity-based Heterogeneous Ordered MARL for VVC

Initialize: Actor networks ϕ_0^i; Global value network ψ, Exeperience Replay buffer $\mathcal{D}$; Initial environment state s; Agents set $\mathcal{N}_a$; Node set $\mathcal{N}$; Number of episodes K; Episode length H
for episode $k = 0, 1 \dots K-1$ **do**
  for time step $t = 0, 1, \dots, H-1$ **do**
    for each agent i_m in $\mathcal{N}_a$ **do**
      $a_{i_m,t} = \pi_{\phi_{i_m}}$, $\boldsymbol{a} = (a_t^{i_1}, \dots, a_t^{N_a})$.
      Calculate rewards r_t using Eq. (17).
    end for
    Calculate V-Q sensitivity matrix $\mathbf{S}_{VQ}$ using (27).
  end for
  Add tuple $\{(s_t^i, a_t^i, s_{t+1}^i, r_t), \forall i \in \mathcal{N}_a, t \in T\}$
  to replay buffer $\mathcal{D}$:
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t^i, a_t^i, s_{t+1}^i, r_t)\}$
  Sample a mini-batch of transitions from $\mathcal{D}$
  Calculate node V-Q sensitivity $S_{VQ}(i)$ by Eq. (28)
  Calculate reward weight $w_{QV,t}$ using Eq. (29)
  Extract the update order $\mathcal{O}_k(i)$ of agents by Eq. (30)
  for each agent i_m in $\mathcal{N}_a$ following order $\mathcal{O}_k(i)$ **do**
    Update the parameter $\phi_{k+1}^{i_m}$ actor network of agent i_m by computing the surrogate function using (24-26)
  end for
  Update critic network by using MSE:
  $\psi_{k+1} = \arg\min_{\psi} \frac{1}{DH} \sum_{d=1}^{\mathcal{D}} \sum_{t=0}^{H} \left(V_\psi(s_t) - \hat{R}_t \right)^2$
end for

The advantage of the order selector is that it prioritizes updating nodes with higher V-Q sensitivity depending on the networks, allowing these sensitive nodes to adjust first. Subse-

quently, lower sensitivity nodes benefit from the adjustments made by high-sensitivity nodes, providing a solid foundation for their control actions. This sequential updating ensures that less sensitive nodes operate based on the improved state established by more sensitive ones, thereby achieving voltage stability more rapidly and enhancing overall control precision. In practice, sensitivity can be calculated using power flow and (27), or directly from historical grid data. Algorithm 1 depicts the SHOM algorithm.

V. Numerical Experiment and Case Studies

The simulations were conducted in Python on a 64-bit server with two 2.30 GHz CPUs and one A800 GPU. To evaluate the proposed method against mainstream algorithms, we a multi-agent RL environment for Volt-Var control called PowerZoo. The code is developed in PyTorch and the power flow calculations are performed by OpenDSSDirect [31], which is a power distribution system simulator package in python. To accelerate the matrix operation used by calculating the sensitivity of nodes, the GPU calculation library CuPy is used [32].

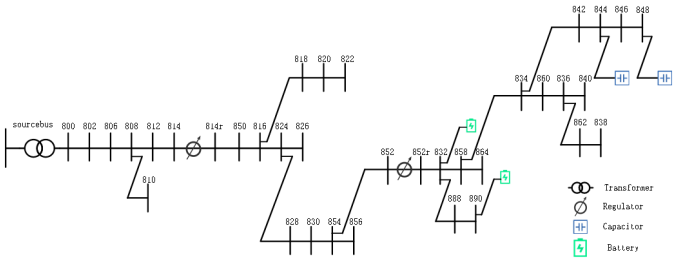


Fig. 5: 34-Bus

A. System Setup and Parameters

The proposed SHOM algorithm is implemented among modified 13-Bus, 34-Bus, and 123-Bus that are adapted as ADNs. Fig. 5 shows a typical 34-Bus diagram. Fig. 6 illustrates a one-day load data slice of two weeks for the load profiles used at 34-Bus network. All load profile and test feeder data are from IEEE PES Test Feeder [33] with hourly basis. To align with the steady-state analysis and the hourly sampling of the load data, the control timescale is also set to one hour per step in the simulation.

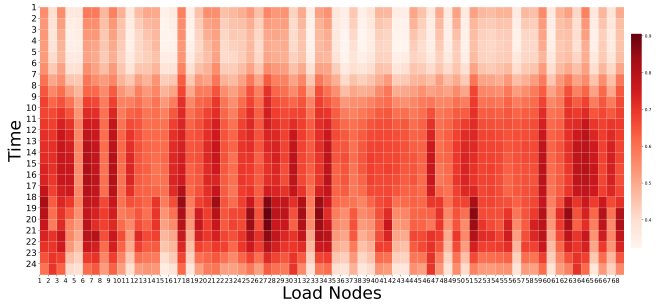


Fig. 6: 34-Bus Load Shapes for One Day

B. Comparison with other MARL Algorithms in 123-Bus

For better illustrates the performance of SHOM in larger network, we compared the algorithms in a 123-Bus network. The average episode rewards for each algorithm is shown in Fig. 7. In the 123-Bus network, SHOM converges to approximately -12.05, outperforming all other MARL algorithms like Qmix and HAPPO. The narrower confidence intervals of SHOM indicate greater stability compared to PPO and SAC, which show larger variations.

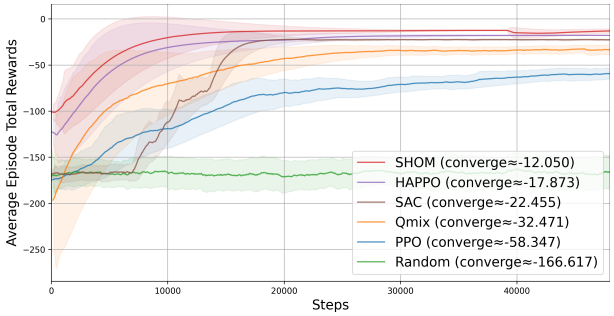
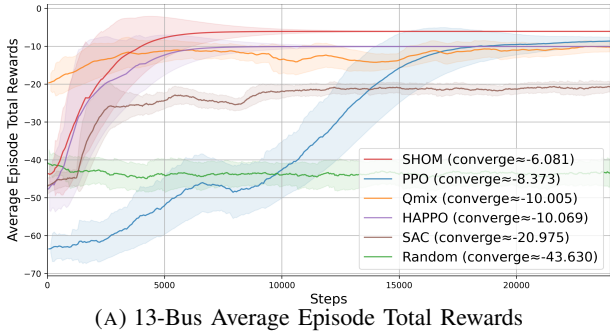
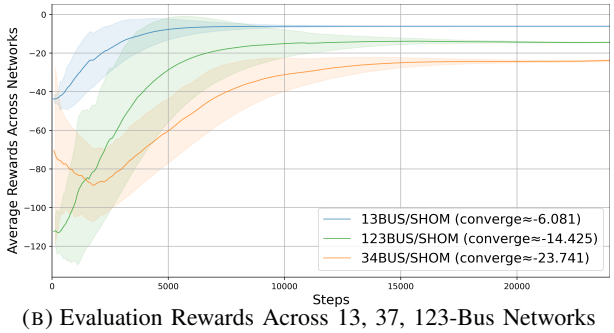


Fig. 7: Overall Comparison within 123-Bus network

C. Overall Performance Across Different Scales of Networks



(A) 13-Bus Average Episode Total Rewards



(B) Evaluation Rewards Across 13, 37, 123-Bus Networks

Fig. 8: Comparison of Evaluation Rewards across Networks and the 13-Bus Network.

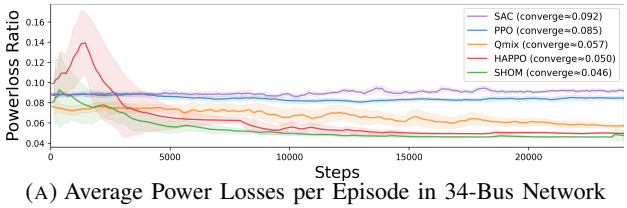
In small-scale networks, we compared centralized RL methods, including PPO and SAC, as well as decentralized methods, including Qmix[34] and HAPPO[28], with SHOM, as shown in Fig. 8a. The training process consists of 24,000 steps across four parallel processes, with each episode comprising 24 steps. For comparison, a random policy was also included. Notably, SHOM converges within approximately 5,000 steps, while other algorithms take significantly longer and achieve lower performance.

Interestingly, we observed that the performance of SHOM varies non-linearly across networks of different scales. As

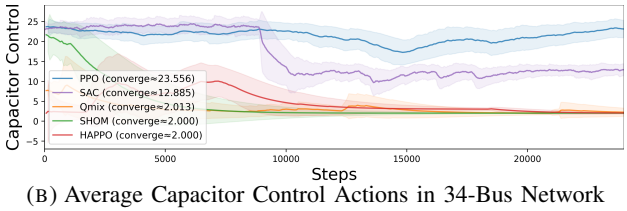
shown in Fig. 8, SHOM performs better in the 123-Bus network compared to the 34-Bus network. This is largely attributed to the 123-Bus network having more regulating devices than the 34-Bus network, thereby offering greater control capacity. Thus, the subsequent discussions primarily focus on the 34-Bus network to highlight the regulating capacity of SHOM under limited resources.

D. Noisy Data Testing

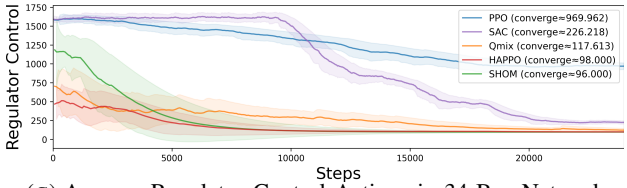
To test the algorithm stability, we further added Gaussian distribution noise with zero mean and a standard deviation of 0.02 p.u. into 34-Bus network, as described in [35, 36]. Furthermore, Fig. 9 illustrates the performance comparison of the algorithms across four key metrics during evaluation. SHOM exhibits superior performance by achieving the lowest average power loss ratio (0.046), minimal control action numbers for capacitors and regulators (2.000 and 96, respectively), and the highest battery utilization (15.863). In contrast, PPO exhibits inefficient performance with the highest number of capacitor and regulator control actions, while SAC shows no battery usage. Qmix and HAPPO both show good convergence rate but still less than SHOM in overall performance.



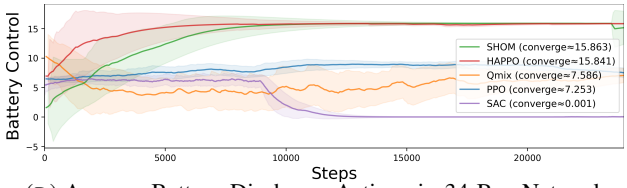
(A) Average Power Losses per Episode in 34-Bus Network



(B) Average Capacitor Control Actions in 34-Bus Network



(C) Average Regulator Control Actions in 34-Bus Network



(D) Average Battery Discharge Actions in 34-Bus Network

FIG. 9: Algorithm Evaluation Metrics in 34-Bus Network

In Fig. 9, SHOM outperforms all other centralized and decentralized algorithms with the fastest convergence rate. The centralized algorithm, such as SAC, fails to utilize the battery effectively for network regulation, as shown in Fig. 9d. Additionally, the performance of each algorithm was compared based on true power loss values, as presented in Fig. 10.

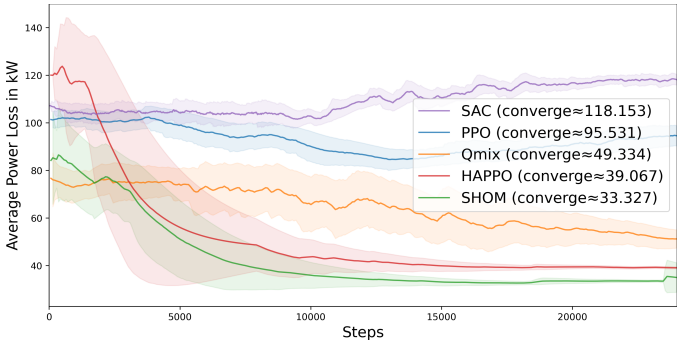


FIG. 10: Noised 34-Bus Power Losses in kW

To provide a clear intuition of how the model outperforms other algorithms and methods in actual deployment, we conducted a deployment experiment using 24 hours of data to evaluate power losses, as depicted in Fig. 11. As shown in Fig. 11, the single-agent algorithm exhibits limited overall regulation capability, resulting in significant power losses. In contrast, the multi-agent algorithm demonstrates superior performance, leading to considerably lower power losses. Additionally, we compared the network losses under a PID controller, represented by the grey bar in the figure. It is noteworthy that the instability of centralized algorithms, such as PPO, can sometimes exceed that of the PID controller. Fig. 12 illustrates the total number of control actions taken

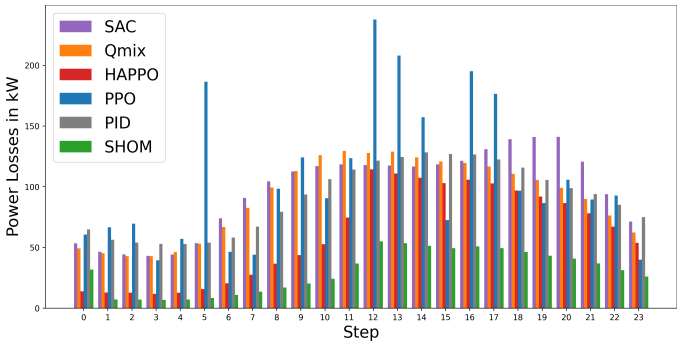


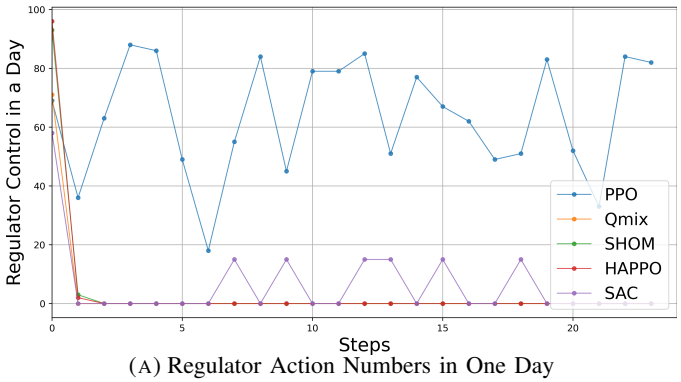
FIG. 11: Power losses in kW within One Day

for the regulators(taps) and capacitors in the 34-Bus network. Interestingly, we found that for SHOM and other MARL algorithms, the regulator and capacitor settings quickly reached an optimal configuration at the beginning, resulting in minimal changes in later times. In contrast, single-agent algorithms like PPO continued to operate the devices throughout the day, which may increase the device wear and lower the lifespan of devices.

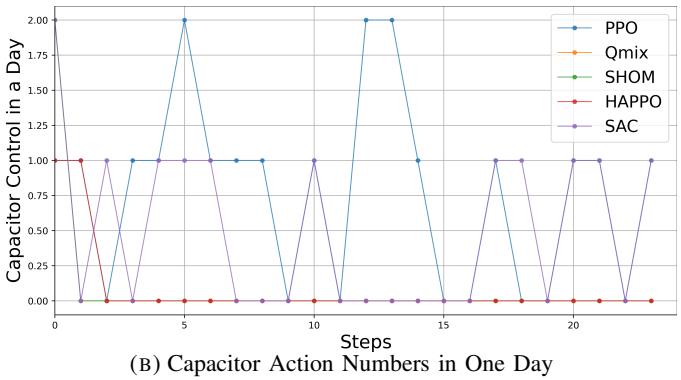
E. Single Node Analysis

In this subsection, we choose BUS-890 in 34-Bus network to analyze the deployment performance of SHOM. By Fig. 5 BUS-890 is connected to a end battery BATT1, which makes it more controllable than other normal buses.

Fig. 13 shows the voltage variation in p.u. over a 24-hour period at BUS 890 in the 34-Bus network. The bus voltage is colored in red, yellow, and green to represent the health status of the bus node. As observed from the graph,



(A) Regulator Action Numbers in One Day



(B) Capacitor Action Numbers in One Day

FIG. 12: Comparison of Regulator and Capacitor Actions in One Day for 34-Bus Network

SHOM maintains the bus voltage close to 1 p.u., despite slight fluctuations during the middle of the day. Qmix is generally capable of controlling the voltage throughout the day, but its average voltage deviation is higher than that of SHOM. In contrast, HAPPO and the centralized algorithms demonstrate poor voltage control ability, with the maximum deviation reaching up to 0.25 p.u.(PPO), which could potentially harm the system. The PID controller could be the benchmark for comparison since it uses a range control policy, which uses fixed thresholds for voltage controlling. However, this methods greatly relies on the local fixed threshold, which would ignore the efficiency of the network and the global stability.

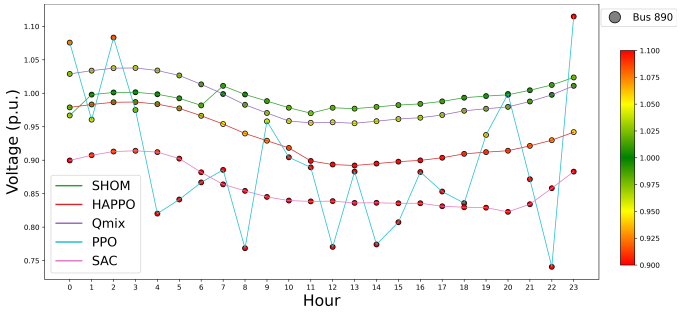
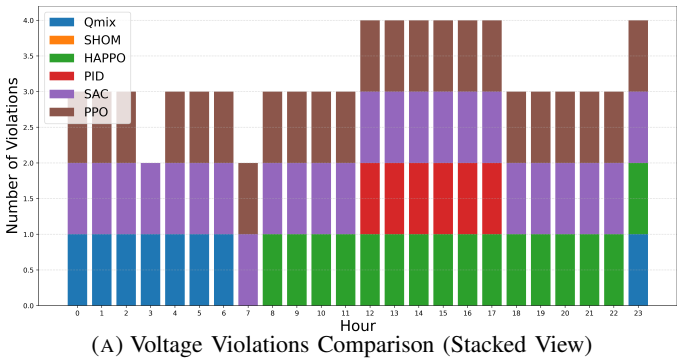
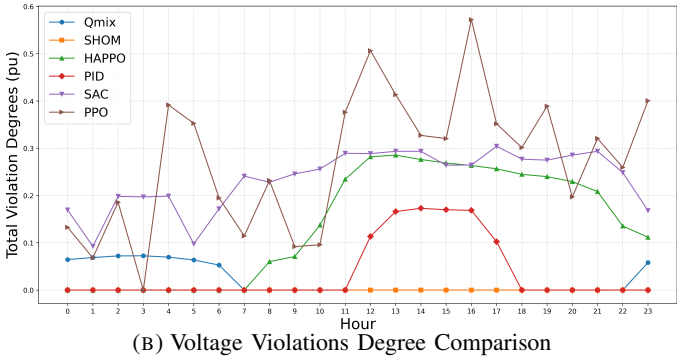


FIG. 13: Bus 890 Voltage Health In a Day

From Fig. 14, the overall voltage violation status within the range of 1 ± 0.05 p.u. for BUS-890 can be observed, including both the count and the absolute extent. Remarkably, under the control of SHOM, BUS-890 experiences no voltage violations throughout the day.



(A) Voltage Violations Comparison (Stacked View)



(B) Voltage Violations Degree Comparison

FIG. 14: Comparison of Voltage Violations

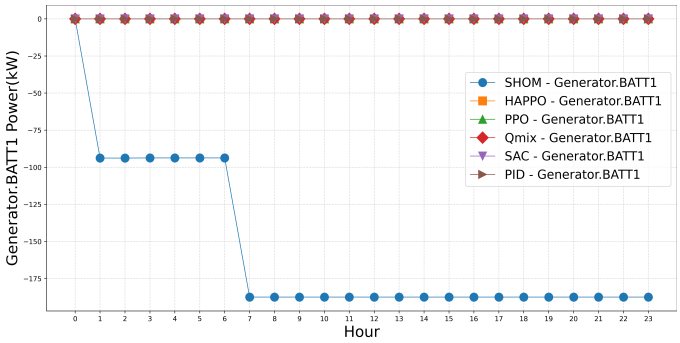
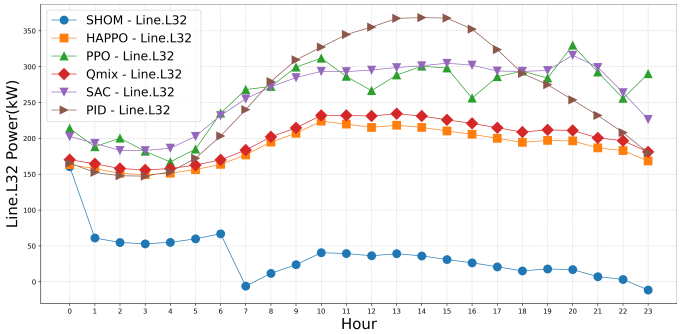


FIG. 15: Battery 1 Power

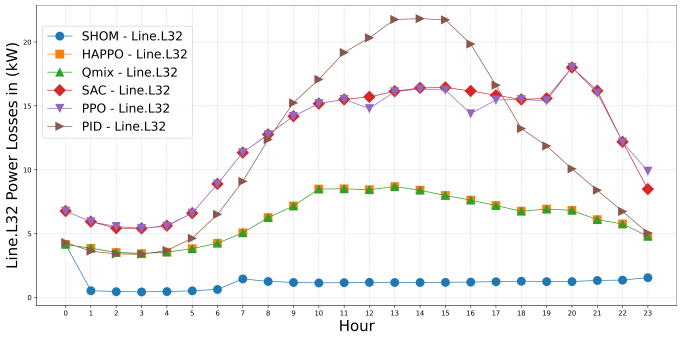
In this paper, the battery is assumed to deliver both active and reactive power, which is typically achieved through a Battery Energy Storage System (BESS). Fig. 15 illustrates how different algorithms utilize the battery system connected to BUS-890. Except for SHOM, all other algorithms fail to fully exploit the regulation capabilities of Battery-1.

Moreover, Fig. 16 depicts the power flow and power losses through Line 32 under the control of SHOM. As we can see from the graph, the powers flow on Line.32 throughout the day is significantly lower than that of other algorithms, which indicates that SHOM effectively utilized the regulating capability of the battery to meet the power demands related to the Bus-890, thereby greatly reducing the power losses on the transmission line, as shown in Fig. 16b

Finally, Table I concludes the 34-Bus network voltage statistics for all buses. It can be seen from Table I that SHOM has significant advantages in voltage regulation compared to other methods. It achieves the lowest percentage of points exceeding



(A) Line 32 Power Flows



(B) Line 32 Power Losses

FIG. 16: Comparison of Line 32 Power Flows and Losses

TABLE I: All Buses Voltage Statistics at 34Bus Network

	SHOM	HAPPO	Qmix	PPO	SAC
Average Excess >1.05 pu	0.0039	0.0170	0.0183	0.0459	0
Maximum Excess >1.05 pu	0.0067	0.0280	0.0554	0.1514	0
Number of <0.95 pu	0	46	0	260	304
Average Excess <0.95 pu	0	0.0156	0	0.0580	0.0352
Maximum Excess <0.95 pu	0	0.0579	0	0.2094	0.1272
Percentage >1.05 pu (%)	2.2522	11.8243	15.0901	19.7072	0
Percentage <0.95 pu (%)	0	5.1802	0	29.2793	34.2342
Line Average Losses(kW)	0.7167	2.1615	1.4357	2.6642	2.3617

1.05 p.u. at 2.25% and maintains no points below 0.95 p.u. Also, SHOM minimizes voltage violation magnitudes with an average excess voltage above 1.05 pu of only 0.0039, and no violations below 0.95 p.u. Furthermore, SHOM achieves the lowest average line power losses at 0.7167kW, which is only 1/3 of other algorithms.

VI. CONCLUSION

This paper proposes a Sensitivity-based Heterogeneous Ordered Multi-agent Reinforcement Learning (SHOM) algorithm to address the Volt-Var Control (VVC) challenge in Active Distribution Networks (ADNs) that include various heterogeneous devices. SHOM leverages multiple agent types working in tandem to control different types of equipment while accounting for individual constraints, thereby significantly improving the adaptability of VVC in real-world ADNs with diverse device types.

To capture the temporal dynamics of power systems, we designed a novel order updating selector based on the reactive power sensitivity. This selector enables the model to learn time-related system characteristics and data features effectively, thereby enhancing its predictive capabilities to dynamic system changes. Case studies on 13-Bus, 34-Bus, and 123-Bus test feeders validate the effectiveness of the proposed SHOM algorithm. Future work may focus on exploring a fully model-free approach by allowing the model itself to determine the updating directions. Such an approach would eliminate the need for calculating sensitivity matrices, thereby significantly reducing computational complexity.

REFERENCES

[1] J. Carrasco, L. Franquelo, J. Bialasiewicz, E. Galvan, R. PortilloGuisado, M. Prats, J. Leon, and N. Moreno-Alfonso, "Power-electronic systems for the grid integration of renewable energy sources: A survey," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1002–1016, Jun. 2006.

[2] H. Liu and W. Wu, "Online multi-agent reinforcement learning for decentralized inverter-based volt-VAR control," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 2980–2990, Jul. 2021.

[3] ——, "Two-stage deep reinforcement learning for inverter-based volt-VAR control in active distribution networks," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2037–2047, May 2021.

[4] X. Yang, H. Liu, and W. Wu, "Attention-enhanced multi-agent reinforcement learning against observation perturbations for distributed volt-VAR control," *IEEE Trans. Smart Grid*, pp. 1–1, 2024.

[5] Y. Nie, J. Liu, X. Liu, Y. Zhao, K. Ren, and C. Chen, "Asynchronous Multi-Agent Reinforcement Learning-Based Framework for Bi-Level Noncooperative Game-Theoretic Demand Response," *IEEE Transactions on Smart Grid*, vol. 15, no. 6, pp. 5622–5637, Nov. 2024.

[6] M. Baran and F. Wu, "Optimal sizing of capacitors placed on a radial distribution system," *IEEE Transactions on Power Delivery*, vol. 4, no. 1, pp. 735–743, Jan. 1989.

[7] L. Gan, N. Li, U. Topcu, and S. H. Low, "Exact convex relaxation of optimal power flow in radial networks," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 72–87, Jan. 2015.

[8] S. H. Low, "Convex relaxation of optimal power flow—part II: Exactness," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 2, pp. 177–189, Jun. 2014.

[9] X. Chen, G. Qu, Y. Tang, S. Low, and N. Li, "Reinforcement learning for selective key applications in power systems: Recent advances and future challenges," *IEEE Trans. Smart Grid*, vol. 13, no. 4, pp. 2935–2958, Jul. 2022.

[10] A. Y. Majid, S. Saaybi, V. Francois-Lavet, R. V. Prasad, and C. Verhoeven, "Deep reinforcement learning versus evolution strategies: A comparative survey," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–19, 2023.

[11] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," in *Proceedings of the 31st International Conference on Machine Learning*, Jan. 2014, pp. 387–395.

[12] H. Li and H. He, "Learning to operate distribution networks with safe deep reinforcement learning," *IEEE*

Trans. Smart Grid, vol. 13, no. 3, pp. 1860–1872, May 2022.

[13] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, Jul. 2017, pp. 22–31.

[14] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, "Two-timescale voltage control in distribution grids using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2313–2323, May 2020.

[15] P. Li, J. Shen, Z. Wu, M. Yin, Y. Dong, and J. Han, "Optimal real-time Voltage/Var control for distribution network: Droop-control based multi-agent deep reinforcement learning," *International Journal of Electrical Power & Energy Systems*, vol. 153, p. 109370, Nov. 2023.

[16] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.

[17] S. Liu, W. Liu, W. Chen, G. Tian, J. Chen, Y. Tong, J. Cao, and Y. Liu, "Learning multi-agent cooperation via considering actions of teammates," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, 2023.

[18] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: A survey," *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 895–943, Feb. 2022.

[19] R. Lowe, YI. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning (ICML)*, Aug. 2018.

[21] X. Sun and J. Qiu, "Two-stage volt/var control in active distribution networks with multi-agent deep reinforcement learning method," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 2903–2912, Jul. 2021.

[22] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. M. de Cote, "A Survey of Learning in Multiagent Environments: Dealing with Non-Stationarity," 2017.

[23] H. Li and H. He, "Multiagent Trust Region Policy Optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2023.

[24] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, Jun. 2015, pp. 1889–1897.

[25] D. Cao, J. Zhao, W. Hu, N. Yu, F. Ding, Q. Huang, and Z. Chen, "Deep reinforcement learning enabled physical-model-free two-timescale voltage control method for active distribution systems," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 149–165, Jan. 2022.

[26] Y. Zhong, J. G. Kuba, X. Feng, S. Hu, J. Ji, and Y. Yang, "Heterogeneous-agent reinforcement learning," *J. Mach. Learn. Res.*, vol. 25, no. 32, pp. 1–67, 2024.

[27] Y. Gao, W. Wang, and N. Yu, "Consensus multi-agent reinforcement learning for volt-VAR control in power distribution networks," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 3594–3604, Jul. 2021.

[28] J. G. Kuba, R. Chen, M. Wen, Y. Wen, F. Sun, J. Wang, and Y. Yang, "Trust region policy optimisation in multi-agent reinforcement learning," Apr. 2022.

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Aug. 2017.

[30] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," 2015.

[31] "OpenDSSDirect.py: A cross-platform Python package that implements a native/direct library interface to the alternative OpenDSS engine from DSS-Extensions.org," Feb. 2024.

[32] R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis, "CuPy: A NumPy-compatible library for NVIDIA GPU calculations," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

[33] K. P. Schneider, B. A. Mather, B. C. Pal, C.-W. Ten, G. J. Shirek, H. Zhu, J. C. Fuller, J. L. R. Pereira, L. F. Ochoa, L. R. de Araujo, R. C. Dugan, S. Matthias, S. Paudyal, T. E. McDermott, and W. Kersting, "Analytic considerations and design basis for the IEEE distribution test feeders," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3181–3188, May 2018.

[34] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, Jul. 2018, pp. 4295–4304.

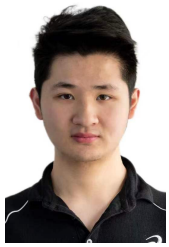
[35] M. Brown, M. Biswal, S. Brahma, S. J. Ranade, and H. Cao, "Characterizing and quantifying noise in PMU data," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*. Boston, MA, USA: IEEE, Jul. 2016, pp. 1–5.

[36] P. Tripathy, S. C. Srivastava, and S. N. Singh, "A Divide-by-Difference-Filter Based Algorithm for Estimation of Generator Rotor Angle Utilizing Synchrophasor Measurements," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 6, pp. 1562–1570, Jun. 2010.

Xiaodong Zheng received the B.S. degree in Electrical Engineering from the Florida Institute of Technology in Melbourne, FL, USA, in 2018, and the M.S. degree in Electrical Engineering from the University of Southern California in Los Angeles, CA, USA, in 2020.

He is currently pursuing the Ph.D. degree in Electrical Engineering with the School of Electrical Engineering at Xi'an Jiaotong University in Xi'an, China. His main fields of interest include multi-agent reinforcement learning, distributed control in power systems, and market-based demand response.

Shixuan Yu received his B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2021. He is currently a postgraduate student at the School of Electrical Engineering at Xi'an Jiaotong University, Xi'an, China. His research interests include control and reinforcement learning and their applications in power systems.



Tianzhuo Shi received the Bachelor's degree in Electrical Engineering from Xi'an Jiaotong University. Currently, he is pursuing a Master's degree in the same field at Xi'an Jiaotong University. Throughout his academic career, he has demonstrated a strong interest in the field of electrical engineering and has accumulated rich knowledge and experience during both his undergraduate and graduate studies.

Shuangsi Xue received the B.E. degree in electrical engineering and automation from Hunan University, Changsha, China, in 2014, and the M.E. and Ph.D. degrees in electrical engineering from Xian Jiaotong University, Xian, China, in 2018 and 2023, respectively.

He is currently an Assistant Professor at the School of Electrical Engineering, Xian Jiaotong University. His current research interest includes adaptive control and data-driven control of networked systems.

Tao Ding Professor of School of Electrical Engineering, Xi'an Jiaotong University, selected as a outstanding young scholar of Shaanxi Province, Elsvier's highly cited scholar in China. He received the B.S.E.E. and M.S.E.E. degrees from Southeast University, Nanjing, China, in 2009 and 2012, respectively, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2015. During 2013 and 2014, he was a Visiting Scholar in the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA. During 2019 and 2020, he was a Visiting Scholar of the Robert W. Galvin Center for Electricity Innovation at Illinois Institute of Technology. His current research interests include electricity markets, power system economics and optimization methods, and power system planning and reliability evaluation. Dr. Ding was funded by 3 NSFC projects, and 30 projects from SG in China. He is an editor of IEEE Transactions on Power Systems, IEEE Transactions on Sustainable Energy, IEEE Power Engineering Letters, IET Generation Transmission & Distribution, and IEEE Systems Journal. He has published more than 300 papers. He received several awards and IEEE PES Outstanding Young Professionals Award.

Hui Cao received the B.E., M.E., and Ph.D. degrees in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2000, 2004, and 2009, respectively. He is a Professor at the School of Electrical Engineering, Xi'an Jiaotong University. He was a Postdoctoral Research Fellow at the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, from 2014 to 2015. He has authored or coauthored over 30 scientific and technical papers in recent years. His current research interest includes knowledge representation and discovery. Dr. Cao was a recipient of the Second Prize of National Technical Invention Award.