Deep Reinforcement Learning Enabled Physical-Model-Free Two-Timescale Voltage Control Method for Active Distribution Systems

Di Cao, Student Member, IEEE, Junbo Zhao, Senior Member, IEEE, Weihao Hu, Senior Member, IEEE, Nanpeng Yu, Senior Member, IEEE, Fei Ding, Senior Member, IEEE, Qi Huang, Senior Member, IEEE, Zhe Chen, Fellow, IEEE

Abstract—Active distribution networks are being challenged by frequent and rapid voltage violations due to renewable energy integration. Conventional model-based voltage control methods rely on accurate parameters of the distribution networks, which are difficult to achieve in practice. This paper proposes a novel physical-model-free two-timescale voltage control framework for active distribution systems. To achieve fast control of PV inverters, the whole network is first partitioned into several sub-networks using voltage-reactive power sensitivity. Then, the scheduling of PV inverters in the multiple sub-networks is formulated as Markov games and solved by a multi-agent soft actor-critic (MASAC) algorithm, where each sub-network is modeled as an intelligent agent. All agents are trained in a centralized manner to learn a coordinated strategy while being executed based on only local information for fast response. For the slower time-scale control, OLTCs and switched capacitors are coordinated by a single agent-based SAC algorithm using the global information with considering control behaviors of the inverters. Particularly, the two-level agents are trained concurrently with information exchange according to the reward signal calculated from the data-driven surrogate model. Comparative tests with different benchmark methods on IEEE 33- and 123-bus systems and 342-node low voltage distribution system demonstrate that the proposed method can effectively mitigate the fast voltage violations and achieve systematical coordination of different voltage regulation assets without the knowledge of accurate system model.

Index Terms—Active distribution systems, coordinated control, deep reinforcement learning, voltage regulation, Volt-Var optimization, PV inverters.

NOMENCLATURE

A. Abbreviations
ADN Active distribution network

PV Photovoltaic
VVC Voltage/var control
OLTC On-load tap changer
SC Switched capacitors
MDP Markov decision process
DRL Deep reinforcement learning
MADRL Multi-agent deep reinforcement learning
DER Distributed energy resources
GP Gaussian process
SAC Soft actor critic
MASAC Multi-agent soft actor critic

B. Parameters

\( G_{ij}/B_{ij} \) The real/imaginary part of admittance element between nodes \( i \) and \( j \)
\( \sigma \) A weighted coefficient between voltage deviation and switching number
\( V_{i,\text{min}}/V_{i,\text{max}} \) Lower/upper bound of voltage of node \( i \)
\( V_{0,i,t,r} \) Reference voltage for OLTC in primary side
\( V_{\text{Tap}} \) Voltage difference between two adjacent tap positions of OLTC
\( S_{\text{PV}} \) Apparent power of PV connected to node \( i \)

C. Variables

\( V_{i,t,r} \) Voltage per unit value of node \( i \) in the time-interval \( \tau \) during time-step \( t \)
\( z_{i,t} \) Total switching numbers of OLTCs and SCs during time-step \( t \)
\( p_{\text{PV},i,t,r}^{PV}/q_{\text{PV},i,t,r}^{PV} \) Active/reactive power injections of PV connected to node \( i \) in time-interval \( \tau \) during time-step \( t \)
\( p_{\text{Load},i,t}^{\text{Load}}/q_{\text{Load},i,t}^{\text{Load}} \) Active and reactive power of load connected to node \( i \) during time-step \( t \)
\( q_{\text{SC},i,t}^{\text{SC}} \) Reactive power injection of SC connected to node \( i \) during time-step \( t \)
\( \theta_{y,i,t,r} \) Voltage phase difference between nodes \( i \) and \( j \) in time-interval \( \tau \) during time-step \( t \)
\( \alpha_{i,j} \) The on-off commitment of OLTC

This work was supported by the National Key Research and Development Program of China (2018YFE0127600). Corresponding author: Weihao Hu.

Di Cao and Weihao Hu are with the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu, China (e-mail: caodi@std.uestc.edu.cn; whu@uestc.edu.cn).

J. Zhao is with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS 39762 (e-mail: junbo@ece.msstate.edu).

Nanpeng Yu is with the Department of Electrical and Computer Engineering at University of California, Riverside. (email: nyu@ece.ucr.edu).

Fei Ding is with National Renewable Energy Laboratory, Golden, CO 80401 USA (e-mail: fei.ding@nrel.gov).

Qi Huang is with the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu, China. He is also with the College of Energy, Chengdu University of Technology, Chengdu, China. (e-mail: hwong@uestc.edu.cn)

Zhe Chen is with the Department of Energy Technology, Aalborg University, Aalborg, Denmark (e-mail: zch@et.aau.dk).
I. INTRODUCTION

With increased penetration of distributed energy resources into the active distribution network (ADN), such as PVs and electric vehicles, the node voltages are subject to more frequent fluctuations and even voltage limit violations. This calls for the development of advanced voltage/var control (VVC) algorithms [1].

Traditional VVC controls are achieved by optimizing the on-load tap changers (OLTCs), switched capacitors (SCs), and voltage regulators [2-3]. It should be noted that traditional VVC control devices are mechanical equipment with slow response speed and limited switching frequencies. For example, these devices are typically operated on an hourly basis [4]. Therefore, control of them is not sufficient to effectively reduce the fast voltage fluctuations caused by the rapid variations of energy sources, such as PVs. By contrast, power electronic devices, such as PV inverters, have much higher response speeds, are promising alternatives for voltage fluctuations mitigation.

According to the type of control hierarchies, the VVC control methods can be divided into centralized and decentralized methods. Centralized methods require global information for decision-making [5-8], which typically require massive calculation processes and complete communication links. Therefore, these methods cannot track the rapid voltage fluctuations caused by the fast variation of PV generations. Decentralized control strategies inform decisions relying on local measurements [9-12]. However, local methods fail to achieve system-wide coordination due to the lack of information exchange. To obtain a balance between the centralized and decentralized control strategies, limited communication links can be utilized to enhance the coordination between different assets [13-16]. Most decentralized methods utilize power electronic devices as voltage regulators to deal with the rapid voltage fluctuations. The coordination of power electronic devices and traditional mechanical devices is not well addressed.

Owing to the different response speeds and characteristics of mechanical and power electronic devices, the joint control of them is a multi-time scale optimization problem. In [17], the two-timescale voltage control is formulated as a two-stage stochastic programming problem. The configurations of PV inverters are pre-determined by the stochastic programming method. This leads to less effectiveness in reducing the rapid voltage fluctuations caused by cloud dynamics. A hybrid control method that combines the advantages of centralized and decentralized control is proposed in [18]. The centralized method is for normal operating conditions. When large fluctuations caused by PVs occur, local control is implemented. A three-stage robust optimization-based method is also proposed for voltage regulation using OLTCs, SCs, and PV inverters [19]. In [20], the scheduling of SCs is formulated as a Markov decision process (MDP) and solved by the deep Q-learning algorithm. The optimal configuration of PV inverters is then transferred to a quadratic program and solved by the commercial solver. In [21], the scheduling of OLTC and SCs is calculated by the optimal power flow method. These settings are then sent to deep reinforcement learning (DRL) agent for the optimization of PV inverters. Although the former methods can provide real-time decisions to reduce voltage fluctuations, the two control hierarchies are not fully coordinated. Therefore,
these methods lack the systematical coordination between the two kinds of assets. [22] proposes a hierarchically coordinated voltage control strategy that can achieve systematic coordination between different assets. A novel distributed coordinated voltage control strategy is proposed for ADN [23]. The aforementioned methods are physical-model-based and thus accurate line parameters and topology are needed [24]-[25]. However, these parameters are typically incomplete and unreliable, especially for secondary feeders [26]-[27].

This paper proposes a physical-model-free two-timescale control framework for the voltage regulation of ADNs utilizing OLTCs, SCs, and PV inverters. The main contributions are:

- **MADRL-enabled coordinated PV inverters to deal with violent voltage fluctuations**: We rely on voltage-reactive power sensitivity to partition the whole network to several sub-regions. Then, each sub-network is modeled as an agent and the coordinated scheduling of PV inverters in multiple sub-networks is formulated as Markov games and solved by the MADRL algorithm with centralized training and decentralized execution. The novel framework allows the proposed method to achieve cooperative control of PV inverters based on only local information and inform fast control decisions to cope with rapid voltage fluctuations caused by rapid variation of PV generations. This differentiates from the classical programming-based two-timescale control methods assuming the pre-determined solutions to deal with the uncertainties. The network partition-based distributed control also allows our proposed method to deal with situations when there is a high proportion of renewable energy generations. This differentiates from our previous work [9] and the method in [21] in the presence of a large number of distributed generators.

- **Physical-model-free control to mitigate assumptions on accurate ADN model and parameters**: This is achieved by developing a new interaction scheme between the surrogate model and DRL agent. Specifically, a surrogate model is first trained in a supervised fashion utilizing historical measurements to replace the original power flow model. This is further embedded into the environment of the DRL algorithm and allows the calculation of reward signals during the training process. This differentiates the proposed method from the physical-model-based approaches [7-9], [13], [17-23] that rely on the accurate network model.

- **Systematetical coordination between fast timescale and slow timescale assets**: For the fast timescale control, the PV inverters optimization is formulated as Markov games and solved by the MASAC algorithm. For the slower timescale control, OLTCs and SCs are coordinated by a single agent-based SAC algorithm using the global information with full consideration of the control behaviors of smart inverters at a faster timescale. The systematical coordination is achieved by training the two-level agents concurrently with information exchange according to the reward signal calculated from the data-driven surrogate model.

The rest of this paper is organized as follows. Section II shows the problem statement. The details of the proposed method are illustrated in Section III. The experimental results are given in Section IV and Section V concludes the paper.

### II. PROBLEM STATEMENT

#### A. Two-timescale Control Framework

![Centralized framework based scheduling of OLTC and SCs](image)

**Centralized framework based scheduling of OLTC and SCs**

![Distributed framework based scheduling of PV inverters](image)

**Distributed framework based scheduling of PV inverters**

In general, the voltage variations are caused by two main factors: the load demand changes and uncertain fluctuations of distributed energy resources (DERs) [1]. The load demand typically changes in a relatively slow and regular pattern, and this is taken care of by traditional VVC devices in the past. However, with the increasing penetration of DERs, conventional VVC devices become less effective in managing fast voltage fluctuations caused by the rapid DER variations. This requires the leverage of fast-responding devices, such as smart PV inverters.

In this context, a two-timescale control should be developed for the joint control of slow-responding mechanical devices and fast-responding PV inverters. Fig. 1 illustrates the two-timescale control problem. Each operational day is divided into $T$ time-steps, each of which is further partitioned to $n$ time intervals. At each time step, a centralized controller calculates the optimal scheduling of OLTCs and SCs based on the global system information. Then, PV inverters inform fast decisions according to only local measurements and the commitments made by OLTCs and SCs at each interval. This two-timescale control can manage the fast voltage fluctuations and achieve effective voltage regulation by the cooperative control of the two kinds of assets.

#### B. Problem Formulation

The objective of the two-timescale voltage regulation is to find the optimal scheduling of OLTCs and SCs at each time step and the reactive power settings of PV inverters at every time interval, such that the total voltage deviation and long-term switching numbers of mechanical devices can be minimized while keeping maximum PV generations. Formally, we have:

$$\min \sum_{i \in M} \sum_{t=1}^{T} \sum_{j=1}^{n} V_{i,j,t} - 1 | + \sigma z_i$$

$s, \mathbf{p}^{pV}_{i,j,t} - \mathbf{p}^{load}_{i,j,t} = V_{i,j,t} \sum_{j=1}^{N} \sum_{t=1}^{n} (G_{i,j} \cos \theta_{i,j,t} + B_{i,j} \sin \theta_{i,j,t}), \forall i \in M$

$$Q^{pc}_{i,j,t} - \mathbf{Q}^{load}_{i,j,t} = V_{i,j,t} \sum_{j=1}^{N} \sum_{t=1}^{n} (G_{i,j} \sin \theta_{i,j,t} - B_{i,j} \cos \theta_{i,j,t}), \forall i \in M$$

$$V_{i,min} \leq V_{i,j,t} \leq V_{i,max}, \forall i \in M$$

$$\alpha_{i,j} \in \{0, 1\}$$

$$\beta_{i} \in \{-10, -9, ..., -1, 0, 1, ..., 9, 10\}, \forall i \in M$$

$$z_{i} = |\alpha_{i,j} - \alpha_{i,j,t}| + |\beta_{i} - \beta_{i,t}|$$

$$Q^{PC}_{i,j,t} = Q^{PC}_{i,j} \forall i \in M$$
where (1) is the objective function; \(V_{i,t} \) represents the voltage per unit value of node \(i\) in the time-interval \(t\) during time-step \(\tau\); \(M\) is the set of buses in the entire system; \(Z_{c}\) denotes the total switching numbers of OLTCs and SCs during time-step \(\tau\); \(\sigma\) is a coefficient to balance the weights between voltage deviation and switching numbers; (2)-(3) are power flow constraints; \(p_{i,t}^{PV}\) and \(Q_{i,t}^{PV}\) represent the active and reactive power injections of PV connected to node \(i\) in time-interval \(\tau\) during time-step \(t\); \(Q_{i,t}^{SC}\) is the reactive power injection of SC connected to node \(i\) during time-step \(t\); \(p_{i,t}^{load}\) and \(Q_{i,t}^{load}\) are the active and reactive power of load connected to node \(i\) during time-step \(t\); \(G_{ij}\) and \(B_{ij}\) are the real and imaginary part of admittance element between nodes \(i\) and \(j\); \(\theta_{i,j,t}\) is the voltage phase difference between nodes \(i\) and \(j\) in time-interval \(\tau\) during time-step \(t\). (4) is the constraint of the voltage for each node, where \(V_{i,min}\) and \(V_{i,max}\) are the lower and upper limits; (5) indicates that the on-off commitment of SC is a binary variable; (6) describes that the commitment of OLTC is an integer that ranges between -10 and 10; (7) calculates the switching numbers of OLTCs and SCs during time-step \(t\); (8) computes the reactive power injections according to the on-off commitment; (9) calculates the voltage of substation based on the position of OLTC; \(V^{Tap}\) denotes the voltage difference between two adjacent positions; (10) is the constraint of the reactive power of PV inverter, where \(S_{i}^{PV}\) is the apparent power of PV connected to node \(i\).

The aforementioned optimization problem is difficult to solve due to the following reasons: i) solving the optimization problem requires the exact values of \(G_{ij}\) and \(B_{ij}\), which are affected by many uncertainties in practice, and are often incomplete and unreliable; ii) the task to solve is a two-level optimization problem, where the decisions of mechanical devices \(\alpha_{i,t}\) and \(\beta_{i}\) as well as PV inverters \(Q_{i,t}^{PV}\) are coupled via the objection function (1) and constraints (2) and (3), and the coordination between the two levels are difficult to achieve; iii) for the multi-stage optimization problem, the decision at each time-step must account for the long term operation times of traditional devices; iv) the discrete variables \(\alpha_{i,t}\) and \(\beta_{i}\) render the optimization problem challenging to solve. To this end, this paper proposes a physical-model-free multi-stage two-timescale control strategy.

**III. PROPOSED MODEL-FREE TWO-TIMESCALE CONTROL FRAMEWORK**

The workflow of the proposed method is illustrated in Fig. 2, where the surrogate model and the interactions with agents at fast and slow timescales are the key components. They will be further elaborated below, followed by the training process of the proposed framework for implementations.

**A. Workflow of the Proposed Control Framework**

The overall workflow is shown in Fig. 2 and it is composed of three components:

- **Surrogate model**: it is trained in a supervised manner to capture the relationship between power injections at each node and statuses of OLTCs and SCs to the node voltage magnitudes.
- **Upper-level agent**: it oversees the schedule of OLTCs and SCs. At each time step, the upper-level agent schedules the OLTCs and SCs according to the global states of the ADN to minimize the voltage deviation and long-term switching numbers of OLTCs and SCs. The decisions from the upper-level agent at each time step are delivered to the lower-level agents for coordinated control of PV inverters in the second stage, which will be demonstrated in Section III-C.
Lower-level agents: they oversee the dispatches of the PV inverters in the second stage. Specifically, the whole network is first partitioned into several sub-networks. Each sub-network is modeled as an intelligent agent, which schedules the PV inverters in its sub-region at each time-interval \( \tau \) according to the real-time regional measurement and the configurations of OLTCs and SCs made by the upper-level agent during the current time step.

For the slower time-scale control, the minimization of voltage deviation and long-term switching numbers of OLTCs and SCs is cast into an MDP, which is then solved by the SAC algorithm; for the fast time-scale control, the dispatch of PV inverters in multiple sub-networks is modeled as Markov games and solved by a MADRL algorithm with centralized training and decentralized execution scheme. To stabilize the training process, the lower level agents are first trained in a centralized manner. Then the two-level agents are trained concurrently with information exchange according to the reward signal calculated by the surrogate model to achieve systematical coordination between different assets.

B. Gaussian Process Regression-Based Surrogate Model

The surrogate model aims to estimate the output nodal voltage \( V_{i,t} \) given input \( (P_{i,t}^{\text{load}}, P_{i,t}^{\text{PV}}, Q_{i,t}^{\text{load}}, Q_{i,t}^{\text{PV}}, \beta_i) \), \( i \in M \). In this study, Gaussian process (GP) regression is selected as the surrogate model to learn the relationship from historical data.

Given training set \((X,Y)\) and test set \((X',Y')\), where \( X \in \mathbb{R}^{N \times C} \) and \( Y \in \mathbb{R}^{N \times d} \) represent the input and output values of the training set, i.e., \((P_{i,t}^{\text{load}}, P_{i,t}^{\text{PV}}, Q_{i,t}^{\text{load}}, Q_{i,t}^{\text{PV}}, \beta_i)\) and nodal voltage magnitudes, respectively; \( X' \in \mathbb{R}^{N' \times C} \) and \( Y' \in \mathbb{R}^{N' \times d} \) are the input and output of the test set. The relationship between input and output can be modeled as \( Y = f(X) + \epsilon \) and \( \epsilon \sim N(0, \sigma^2) \) is the homoscedastic Gaussian noise: \( f(X) \) is the Gaussian process mapping from the input to output, which is specified by

\[
f(X) \sim \text{GP}(m(X), K(X,X') + \sigma^2 I) \tag{11}
\]

where \( m(X) \) is the mean value and \( K(X,X') \) is the covariance function; \( K(\cdot) \) is the kernel function; \( I \) is an \( N \times N \) identity matrix. GP aims to forecast \( f(X') \) given new input \( X' \). According to [29], the posterior distribution of \( f(X') \) can be obtained as:

\[
f(X') | Y; X, X' \sim N(\mu', \Sigma') \tag{12}
\]

\[
\mu' = K(X',X) (K(X,X) + \sigma^2 I)^{-1} Y \tag{13}
\]

\[
\Sigma' = K(X',X') - K(X',X) (K(X,X) + \sigma^2 I)^{-1} K(X,X') \tag{14}
\]

where \( \mu' \) and \( \Sigma' \) represent the mean and covariance of new samples. The parameters to be optimized during the training of GP are collected in \( \beta \), which can be updated by maximizing the negative log-likelihood function \(- \log p(Y|X)\) [29]. When the optimal parameters are obtained, the nodal voltages can be estimated by surrogate model according to (13) and (14).

C. Slow Time-scale Control via Soft Actor-critic Algorithm

1) Model the Slow Time-scale Problem As an MDP

The scheduling of OLTCs and SCs is formulated as an MDP, whose main components are as follows [30]:

**State space:** \( S_t \) is the state space of the upper-level agent. The state at \( t \) \( s_t \in S_t \) is defined as \( s_t = (r_t^{\text{load}}, P_t^{\text{load}}, r_t^{\text{PV}}, \alpha_{t-1}, \beta_{t-1}) \), where \( P_t^{\text{load}} \) and \( Q_t^{\text{load}} \) represent real and reactive power of load demand for each node at time-step \( t \), respectively; \( \alpha_{t-1} \) denotes the position of SCs at time-step \( (t-1) \); \( \beta_{t-1} \) denotes the tap positions of OLTCs at time-step \( (t-1) \).

**Action space:** \( A_t \) represents the action space of the upper-level agent. The action at time-step \( t \), \( a_t \in A_t \) is defined as \( a_t = (\alpha_t, \beta_t) \), where \( \alpha_t \) and \( \beta_t \) denote the on/off positions of the SCs and the tap positions of the OLTCs at the current time-step.

**Reward function:** \( R_t \) is the reward function of the upper-level agent and it is defined as the sum of nodal voltage deviation of the whole network and switching numbers:

\[
r_t = -(\sum_{i=1}^{N} |V_{i,t} - 1| + \sigma z_t) - \eta, \tag{15}
\]

where \( \eta \) represents the penalty term when voltages cross the limit.

At each time step, the agent obtains a global observation of the ADN \( S_t \), based on which it makes decisions \( a_t \). Then, the action \( a_t \) is delivered to the fast time-scale devices according to which the lower-level agents inform decisions for \( n \) time intervals. After that, the upper-level agent obtains a reward \( r_t \), and the system transfers to the next state \( s_{t+1} \). The agent aims to learn a policy \( \pi(a_t | s_t) \) to maximize the discounted cumulative reward \( \gamma_t = \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{n} r_{t+n} \) obtained by the agent from the current time-step onward, where \( \gamma \in [0, 1] \) is the discount factor.

2) Solve the MDP via the SAC Method

The optimization of slow time-scale devices is solved by the SAC algorithm, which is composed of actor and critic functions. The actor functions map from \( S_t \) to \( A_t \) and the critic functions provide a judgment of the decision made by the actor function. The actor and critic functions are trained against each other for the formulation of the control strategy.

The critic function \( Q^\pi \), which is also named as action-value function, is utilized to provide gradient information for learning the control strategy. The critic function takes \((s_t, a_t)\) as input and outputs a judgment of the value of action \( a_t \) under the current state \( s_t \). The parameters of the critic functions are optimized by minimizing the following loss function:

\[
L = (y_t - Q^\pi(s_t, a_t))^2 \tag{15}
\]

\[
y_t = r_t + \gamma Q^\pi(s_{t+1}, a_{t+1}) \tag{16}
\]

where \( Y_t \) represents the target value. The target functions are typically introduced to stabilize the training process by replacing \( Q^\pi(\cdot) \) with the target critic function \( Q^{\pi^*}(\cdot) \) in (16). In addition, an entropy term is incorporated to encourage the exploration ability of the algorithm, then (16) is rewritten as:

\[
y_t = r_t + \gamma E_{\alpha_{t+1}}[\log p(\pi^*(a_{t+1} | s_{t+1})) + Q^{\pi^*}(s_{t+1}, a_{t+1})] \tag{17}
\]
where $\delta$ is the temperature parameter to balance the weight between the entropy term $\log(\pi^\theta(a_{i,t}|s_{i,t}))$ and $Q^\theta(s_{i,t},a_{i,t})$.

The actor function, which is also named the policy function, aims to learn a function mapping $s_{i,t}$ to $a_{i,t}$ so as to maximize the output of the action-value function. The actor function is optimized according to the policy gradient:

$$\nabla_{\mu} J(\mu) = E_{s_{i,t},a_{i,t}}[\nabla_{\mu} \log(\pi^\theta(a_{i,t}|s_{i,t}))Q^\theta(s_{i,t},a_{i,t})]$$

Since the entropy term is introduced to avoid convergence to sub-optimal solutions and improve the exploration capability of the algorithm, (18) is rewritten as [31]:

$$\nabla_{\mu} J(\mu) = E_{s_{i,t},a_{i,t}}[\nabla_{\mu} \log(\pi^\theta(a_{i,t}|s_{i,t}))\rho(s_{i,t},a_{i,t})]$$

$$\rho(s_{i,t},a_{i,t}) = -\delta \log(\pi^\theta(a_{i,t}|s_{i,t})) + Q^\theta(s_{i,t},a_{i,t}) - E_{a_{i,t} \sim \pi^\theta}[Q^\theta(s_{i,t},a_{i,t})]$$

where $E_{a_{i,t} \sim \pi^\theta}[Q^\theta(s_{i,t},a_{i,t})]$ is a baseline term, indicating the average state-action value obtained under the current state. $Q^\theta(s_{i,t},a_{i,t}) - E_{a_{i,t} \sim \pi^\theta}[Q^\theta(s_{i,t},a_{i,t})]$ denotes the advantage of the current action over the average value.

Thanks to the strong nonlinear fitting capability of the neural network, it is selected to approximate the critic function in this study. Then, the optimization of parameters for critic and actor functions transforms to the update of the parameters of critic networks $\theta^C$ and actor networks $\theta^A$, respectively. However, the data generated by the agent during the interaction with the environment are typically non-stationary and highly correlated, which may cause divergence during the training of neural networks. To this end, the replay buffer mechanism is applied to reduce the correlation between the training data. At each time step, the transition experience $(s_{i,t},a_{i,t},r_{i,t},s_{i,t+1})$ is stored in the memory of the agent. When the number of transitions reaches a certain amount, a batch of transition data is sampled from the memory at each time step for the calculation of the gradient. Then, the parameters of the critic networks are updated according to the gradient rule:

$$L(\theta^C) = \frac{1}{B} \sum_{i=1}^{B} (y_i - Q^\theta(s_{i,t},a_{i,t}))^2$$

$$\theta^C \leftarrow \theta^C + \eta^C \nabla_{\theta^C} L(\theta^C)$$

The parameters of the actor-networks are optimized via

$$\nabla_{\mu} J(\theta^A) = E_{s_{i,t},a_{i,t}}[\nabla_{\mu} \log(\pi^\theta(a_{i,t}|s_{i,t}))\rho(s_{i,t},a_{i,t})]$$

$$\theta^A \leftarrow \theta^A + \eta^A \nabla_{\theta^A} J(\theta^A)$$

For details of ordinal encoding for discrete actions, the reader can refer to [3].

D. Fast Time-scale Control via MADRL

1) Network Partition

The objective of the network partition is to divide the whole network into several sub-regions such that the centralized optimization problem is decomposed into several small sub-problems that can be solved in a decentralized manner. Since the fast time-scale control is to reduce voltage deviations utilizing PV inverters, an electrical distance based on voltage-reactive power sensitivity is first calculated. Then, the spectral clustering algorithm is applied to search for the optimal partition results of ADN. For more details about the network partition, please refer to our previous work [13].

2) Model the Fast Time-scale Problem As a Markov Game

The scheduling of PV inverters in multiple sub-networks is formulated as Markov games, which is a multi-agent extension of MDP. In the Markov games, each sub-network is modeled as an agent to schedule PV inverters in its sub-region. The main components of the Markov game are as follows:

State set: $S_{i,t}$ represents the state set of all agents in time-interval $t$ during time-step $t$ the state of agent $i$ at time-interval $\tau$ during time-step $t$, $s_{i,t} \in S_{i,t}$ is composed of $(p_{i,t},Q_{i,t},\rho_{i,t},\alpha_i,\beta_i)$, where $i$ represents the index of nodes that are located in sub-network $i$.

Action set: $A_{i,t}$ denotes the action set of all agents in time-interval $t$ during time-step $t$. The action of agent $i$ at time-interval $\tau$ during time-step $t$, $a_{i,t} \in A_{i,t}$ is defined as $a_{i,t} = \lambda_{i,t}$, based on which the reactive power of the PV inverters located in sub-network $i$ can be obtained:

$$Q_{i,t} = \lambda_{i,t}\sqrt{(S_{i,t})^3 - (P_{i,t})^3}$$

Reward function: the reward obtained by agent $i$ at time-interval $\tau$ during time-step $t$ is the sum of voltage deviation of the system: $r_{i,t} = -\sum_{j \in \mathcal{N}}\|V_{i,t} - 1\| - \eta$, which is calculated via (13).

At each time interval, each agent informs decision $a_{i,t}$ according to its observation $s_{i,t}$, and obtains a reward $r_{i,t}$ that is calculated based on global state $S_{i,t}$ and action $A_{i,t}$. Then, the system transfers to the next state $S_{i,t+1}$. Each agent aims to learn a policy $\pi_i(a_{i,t}|s_{i,t})$ to maximize its obtained reward.

3) Solve the Markov Game via the MASAC Algorithm

The MASAC algorithm is utilized to solve the Markov game. Specifically, each PV inverter is modeled as a SAC agent, including the local actor functions and global critic function. All agents are trained in a centralized manner to learn a coordinated control strategy via the global critic function and inform fast decentralized decisions based on local information.

The critic function of each agent $Q^\pi(\cdot)$ takes the global observation $S_{i,t}$ as input and informs actions of all agents to provide a better judgment of the action made by the agent under the current state. The parameters of the critic function of agent $i$ are updated by minimizing the following loss [32]:

$$L = (y_{i,t} - Q^\pi(S_{i,t},a_{i,t},...,a_{N,t}))^2$$

$$y_{i,t} = r_{i,t} + \gamma E_{a_{i,t+1}}[\rho(s_{i,t},a_{i,t},...)Q^\pi(S_{i,t+1},a_{i,t+1},...,a_{N,t+1})]$$

The actor of each agent $\pi^A(\cdot)$ maps from its local observation $S_{i,t}$ to $a_{i,t}$, the parameters of which are optimized according to the policy gradient [30]:

$$\nabla_{\mu} J(\mu) = E_{s_{i,t},a_{i,t}}[\nabla_{\mu} \log(\pi^A(\alpha_{i,t}|\tau,s_{i,t}))\rho(s_{i,t},a_{i,t},...,a_{N,t})]$$

$$\rho(s_{i,t},a_{i,t},...,a_{N,t}) = -\delta \log(\pi^\theta(a_{i,t}|s_{i,t})) + Q^\theta(s_{i,t},a_{i,t},...,a_{N,t}) - E_{a_{i,t} \sim \pi^\theta}[Q^\theta(s_{i,t},a_{i,t},...,a_{N,t})]$$

$$b(s_{i,t},a_{i,t}) = E_{s_{i,t},a_{i,t}}[Q^\pi(S_{i,t},a_{i,t},...,a_{N,t})]$$

where $b(s_{i,t},a_{i,t})$ is the baseline term, representing the average value obtained under current state $S_{i,t}$ when different actions
are taken; \( Q'(S_t, a_t, \ldots, a_{t+\tau}) - b(S_t, a_t, \ldots, a_{t+\tau}) \) is the advantage value, indicating the advantage of current action over the average value; (27) denotes that policy optimization aims to maximize the entropy term and advantage value by optimizing the parameters of actor function.

The neural network and experience replay mechanism are also introduced. The parameters of the critic functions of agent \( i \) are transformed to the parameters of critic networks \( \theta^\pi_i \), which are updated according to the gradient rule:

\[
L(\theta^\pi_i) = \frac{1}{B} \sum_{i=1}^{B} \left(y_i - Q'(S_t, a_t, \ldots, a_{t+\tau})\right)^2
\]

(30)

\[
\theta^\pi_i \leftarrow \theta^\pi_i + \eta_i \nabla \theta^\pi_i L(\theta^\pi_i)
\]

(31)

The parameters of the actor function of agent \( i \) are transformed to the parameters of actor networks \( \theta^\mu_i \) that are optimized via

\[
\nabla \theta^\mu_i J(\theta^\mu_i) = E_{\pi_i \sim \pi}(\nabla \pi_i(\theta^\mu_i | S_t) \rho_i(S_t, a_t, \ldots, a_{t+\tau}))
\]

(32)

\[
\theta^\mu_i \leftarrow \theta^\mu_i + \eta_i \nabla \theta^\mu_i J(\theta^\mu_i)
\]

(33)

E. Training of the Proposed Physical-model-free Two Time-scale Control Method

The parameter set to be optimized for the proposed method is denoted as \( \theta = \{\theta_1, \theta_2, \theta_3\} \), where \( \theta_1 = \{\sigma_1^2, M, \sigma_2^2\} \) is the parameter set for the surrogate model, \( \theta_2 = \{\theta^\pi, \theta^\mu\} \) and \( \theta_3 = \{\theta_1^\pi, \theta_2^\pi, \ldots, \theta_n^\pi, \theta_N^\pi\} \) are the parameter sets of the upper-level and lower-level agents. The surrogate model is first trained in a supervised fashion. Then, it is integrated with the proposed control model, which can develop optimal control behavior by continuous interaction with the surrogate model. The detailed training procedures are as follows:

1) Centralized Training of the Lower Level Agents

The lower level agents are first trained in a centralized manner to enhance the coordination of PV inverters. The details of the training procedure are shown in Algorithm 1. In the beginning, the parameters of the lower-level agents \( \theta_i \) are randomly initialized. The parameters of target networks \( \theta_1^\mu \) and \( \theta_2^\mu \) are copied from the online ones. Then, the parameters are optimized for \( M_1 \) epochs in the outer loop. Each epoch represents an operation day randomly sampled from the training set. An operation day is composed of \( T \) time-steps, each of which is consisted of \( N \) time intervals. At the beginning of each time-step, the status of OLTC and SCs are randomly selected. Then, these information are delivered to lower level agents, which inform decisions according to the status of mechanical devices and the its local observation \( s_{i,t}\). Note that the action of each lower agent is randomly selected to explore the action space at the early stage of the training. When the replay buffer reaches its capacity, each agent calculates its actions according to \( a_{i,t} = \pi^\mu_i(s_{i,t}) \). When actions of all agents are executed, each agent receives an immediate reward \( r_t \), and the system transfers to the next state \( s_{i,t+1} \). After that, the experienced pairs \( (s_{i,t}, a_{i,t}, r_t, s_{i,t+1}) \) are stored in the memory of each agent.

Algorithm 1: Centralized training of the lower level agents

When \( n \) time-intervals are finished, the upper-level agent obtains a reward \( r_t = \sum_{i=1}^{N} r_{i,t} + \sigma_{\xi_i} \), and the system transfers to the next state \( s_{i,t}' \). Next, the experiences \( (s_{i,t}' , a_{i,t}' , s_{i,t+1}') \) are stored in its replay buffer.

When the capacity reaches the upper limit, a batch of experiences is randomly sampled from the memory for each lower level agent. Then, the parameters of critic networks of each lower-level agent \( \theta_1^\mu \) are optimized according to (30) and (31), and the parameters of actor-networks \( \theta_2^\mu \) are optimized according to (32) and (33). The parameters of target networks are optimized according to the following soft update rule:

\[
\theta_1^\mu \leftarrow \pi^\mu_i + (1-\tau)\theta_1^\mu \quad \theta_2^\mu \leftarrow \pi^\mu_i + (1-\tau)\theta_2^\mu
\]

(34)

where \( \tau \ll 1 \) denotes the soft update coefficient.

2) Concurrent Training of the Two Level Agents
At this stage, the upper and lower level agents are trained concurrently to achieve systematical coordination of the two-timescale devices. The details of the concurrent training procedure are shown in Algorithm II. At the beginning, the parameters of the lower level agents are loaded and parameters of upper level agent are randomly initialized. The experience data stored in the replay buffer of each agent are also loaded. Then the two level agents are trained for $M_2$ epochs for the formulation of a coordinated control strategy. At the beginning of each time-step, the upper-level agent obtains a global observation of the distribution network $s_i$, based on which it makes decisions according to $a_i = \pi^\theta(s_i)$. The status of mechanical devices are delivered to the lower level agents, which make decisions according to $a_{i,t} = \pi^{\theta'}(s_{i,t})$. Then, actions $a_i$ and $a_{i,t}$ are executed, and a reward $r_{i,t}$ is calculated by the surrogate model is obtained by each lower agent. After that, each lower level agent stores the new experience in its replay buffer.

When $n$ time-intervals are finished, the upper-level agent stores new experience in its memory. Then, a random batch of experiences is sampled from the replay buffer for the optimization of neural networks, where the parameters of the critic networks are updated according to (21) and (22), and parameters of actor-networks are optimized according to (23) and (24). The parameters of target networks are optimized by

$$
\theta^\mu \leftarrow r\theta^\mu + (1-r)\theta^\mu, \ \ \ \ \ \theta^\mu \leftarrow r\theta^\mu + (1-r)\theta^\mu
$$

(35)

The parameters of lower level agents are then optimized according to (30)-(34).

Algorithm II: Concurrent training of the two level agents

- Randomly initialize parameters of upper level agent $\theta^\mu$, load parameters of lower level agents $\theta^\mu_i$
- Load memories of the upper and lower agents
- Set epoch to 0
- If epoch <= $M_1$
  - Yes
  - Epoch + 1
- Set time-step $t$ to 0
- If time-step $t$ <= $T$
  - No
  - Yes
  - Upper level agent obtains an observation of the environment $s_i$, and makes decision according to $a_i = \pi^\mu (s_i)$
  - Set time-interval $\tau$ to 0
  - No
  - Yes
  - Obtain observation $s_{i,t}$, and calculate action according to $a_{i,t} = \pi^{\mu'} (s_{i,t})$ for each lower level agent $i$
  - Execute actions $a_i$ and $a_{i,t}$ of each agent $i$, calculate reward $r_{i,t}$ based on (13) and (14) and obtain new observation $s_{i,t+1}$ for each lower level agent $i$
  - Store the experience $(s_{i,t}, a_{i,t}, r_{i,t}, s_{i,t+1})$ in memory for each lower level agent $i$
  - Upper level agent obtains reward $r = \sum s_i + \sigma y$, and new observation $s_i$, stores the experience $(s_i, a_i, r_i, s_{i+1})$ in memory
  - Update parameters of upper level agent according to (21)-(24) and (35), update parameters of lower level agents according to (30)-(34)
- $t+1$

Algorithm III: Real-time Implementation of the Proposed Method

- $t+1$
- If time-interval $\tau$ <= $n$
  - No
  - Yes
  - Obtain observation $s_{i,t}$ and calculate decision according to $a_{i,t} = \pi^{\mu'} (s_{i,t})$ for each upper level agent
  - Set time-interval $\tau$ to 0
  - $t+1$
- If time-interval $\tau$ <= $n$
  - No
  - Yes
  - Obtain observation $s_{i,t}$ and calculate action according to $a_{i,t} = \pi^{\mu'} (s_{i,t})$ for each lower level agent $i$
  - Output the concatenated actions of all agents $(a_1, a_2, ..., a_{l,l})$
- End

F. Real-time Implementation of the Proposed Method

When the training process is completed, the parameters of neural networks are fixed and only the actor-network of each agent is kept for the scheduling of controllable devices [33]. The flowchart of the real-time scheduling algorithm is shown in Algorithm III.

In Algorithm III, the parameters of the actor-networks for the upper-level agent $\theta^\mu$ and lower-level agents $\{\theta^{\mu'}_i\}$ are first loaded. At the beginning of each time-step, the upper-level agent calculates action $a_i$ according to $a_i = \pi^{\mu'} (s_i)$. Then, the decisions of mechanical devices are delivered to the lower-level agents, which inform decisions according to $a_{i,t} = \pi^{\mu'} (s_{i,t})$ at each time-step. After that, the actions of all agents are concatenated and executed. Since the lower-level agents make decisions based on only local information and the status of traditional devices, which are delivered by the upper level agent in advance, they can provide fast control decisions.

Remark: Since the optimal configuration made by the upper-level agent is delivered to the lower agents during the training process, the impacts of the slow time-scale devices on the PV inverters are fully considered. In addition, the reward of the upper-level agent is calculated based on the sum of reward obtained by the lower-level agents, which encourages the upper agent to consider the possible compensating capability of PV inverters during the concurrent training process. The information exchange and reward design during the concurrent training procedure enable the proposed method to achieve systematical coordination between the different kinds of assets.
The centralized training and decentralized execution framework helps enhance the coordination between the lower-level agents even only local measurements are utilized. Historical measurement data are first utilized to train a surrogate model, which is applied to provide the reward signal during the training of the DRL agents. This novel interaction scheme allows it to achieve physical-model-free control.

IV. CASE STUDY

In this section, comparative tests are carried out on IEEE 33-, 123-, and 342-node systems to evaluate the performance of the proposed method. The detailed parameter settings are first provided, followed by the performance evaluation of the surrogate model and control model.

A. Simulation Setup

![Topology and subsystem partition results of the test systems](image)

The simulations will be first carried out on the IEEE 33-bus distribution system [34], the topology and subsystem partition results [13] of which are shown in Fig. 3. One OLTC, three SCs, and six PV inverters are used for voltage regulation. The OLTC is connected to node 1. It has 21 tap positions, which correspond to turn ratios between 0.95 and 1.05. There are six PVs at nodes 5, 12, 27, 50, 65, 76, 81, 100, 114 and 118. The active power capacities of PVs are 1.2 MW, and the nominal power of PV inverters are 1.26 MVA.

Further tests are carried out on IEEE 342-node low voltage networked test system [34]. There are 48 PV and 6 SCs utilized for voltage regulation of the system. The active power capacity of PVs are 500 kW, and the nominal power of PV inverters are 550 kVA. The capacity of SCs is set to 300 kVar. The whole network is partitioned to six sub-networks according to responsibility region. There are six lower level agents in total, each corresponding to one sub-region and in charge of eight PV inverters in its region.

The lower and upper bounds of node voltages are 0.95 and 1.05, respectively. For the training of the surrogate model, numerous instances of the input data \( \{P_{i,f}, P_{PV, i,f}, Q_{PV, i,f}, Q_{load, i,f}, Q_{sc, i,f}, \beta_i\}, i \in M \) are first generated, where the load data are multiplied with the random coefficient extracted from [36]. The PV generations are from the field measurements of Xiaojin, a county in the Sichuan province of China. The actions of different controllable devices are randomly generated. Then, the nodal voltages \( \{V_{i,t}\}, i \in M \) are calculated by the power flow model. The numbers of instances for training the surrogate models of the three systems are set to 300, 2000, and 2500, respectively. Each training sample consists of two components: the input \( X = \{P_{i,f}, P_{PV, i,f}, Q_{PV, i,f}, Q_{load, i,f}, Q_{sc, i,f}, \beta_i\}, i \in M \) and \( Y = \{V_{i,t}\}, i \in M \). The features of the training data are listed in Table I.

<table>
<thead>
<tr>
<th>Test system</th>
<th>Input</th>
<th>Output</th>
<th>Training instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>33-node</td>
<td>P_{i,f}, P_{PV, i,f}, Q_{PV, i,f}, Q_{load, i,f}, Q_{sc, i,f}, \beta_i</td>
<td>V_{i,t}</td>
<td>300</td>
</tr>
<tr>
<td>123-node</td>
<td>P_{i,f}, P_{PV, i,f}, Q_{PV, i,f}, Q_{load, i,f}, Q_{sc, i,f}, \beta_i</td>
<td>V_{i,t}</td>
<td>2000</td>
</tr>
<tr>
<td>342-node</td>
<td>P_{i,f}, P_{PV, i,f}, Q_{PV, i,f}, Q_{load, i,f}, Q_{sc, i,f}, \beta_i</td>
<td>V_{i,t}</td>
<td>2500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAC</td>
<td>32</td>
</tr>
<tr>
<td>MASAC</td>
<td>256</td>
</tr>
<tr>
<td>Batch size</td>
<td>2.4e3</td>
</tr>
<tr>
<td>Memory size</td>
<td>10e6</td>
</tr>
<tr>
<td>Temperature parameter</td>
<td>1.25e-3</td>
</tr>
<tr>
<td>Soft update coefficient</td>
<td>0.001</td>
</tr>
<tr>
<td>Learning rate for actor network</td>
<td>0.001</td>
</tr>
<tr>
<td>Learning rate for critic network</td>
<td>0.001</td>
</tr>
<tr>
<td>Neuron number of hidden layer</td>
<td>100/100/100</td>
</tr>
</tbody>
</table>

For training of the DRL-based control model, the PV generation data in Xiaojin are utilized. The PV generation data are divided into the training set and test set, which contains 300- and 10-days’ data, respectively. The load data of each node are composed of three components: the baseline, the temperature, and the random coefficient. During the training procedure, 6000 sets of random coefficients of each node are randomly sampled from 0.8 to 1.2 [35]. When the training process is completed, 240 sets of new random coefficients are randomly generated from this range for the test procedure. The controlling decisions include the on/off positions of the SCs \( \alpha_i \), the tap positions of the OLTCs \( \beta_i \), and \( \lambda_{i,t} \) that is utilized to calculate the reactive power of the \( i \)th PV inverter.
Square exponential covariance is selected as the kernel function for the GP method. The settings of hyper-parameters of the control model for both systems are shown in Table II. Each time-step and time interval corresponds to 1 hour and 15 minutes in this study, respectively. Note that when finishing the training procedure, the trained agents can deal with the scenario with different time-step and time intervals. The proposed method is implemented in Python using Tensorflow and all case studies are carried out on a computer with a 3.0 GHz Intel Core i9-10980XE CPU and 32 GB RAM.

### B. Evaluation of the Proposed Surrogate Model

Table III MAE achieved by various surrogate models on several test systems

<table>
<thead>
<tr>
<th>Method</th>
<th>33-node</th>
<th>123-node</th>
<th>342-node</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLR</td>
<td>1.48e-3</td>
<td>2.27e-3</td>
<td>-</td>
</tr>
<tr>
<td>BPNN</td>
<td>4.6e-4</td>
<td>5.6e-4</td>
<td>1.53e-3</td>
</tr>
<tr>
<td>Proposed</td>
<td>3.2e-4</td>
<td>4.6e-4</td>
<td>5.29e-4</td>
</tr>
</tbody>
</table>

The mean absolute error (MAE) of the voltages estimated by surrogate models and those calculated by the power flow using accurate line parameters are shown in Table III. It can be found that the MLR method has the worst performance on IEEE 33- and 123-bus systems. It fails to learn a good surrogate model of IEEE 342-node system utilizing 2500 instances of training samples. Due to the strong nonlinear fitting ability, BPNN achieves better results than MLR. The proposed method further improve the forecasting accuracy. It outperforms BPNN method by 30.4%, 17.9%, and 65.4% on IEEE 33-, 123-, and 342-node system, respectively. The distributions of errors between the estimated voltages by different surrogate models and the actual voltages calculated by the power flow on three test systems are shown in Fig. 4. It can be observed that the error distribution of the proposed method gets closer to 0 compared with the MLR and BPNN methods. In addition, the proposed method has the lowest maximum error in all the test systems.

### C. Evaluation of the Proposed Control Model

Comparative tests are carried out on 10 days’ test data to evaluate the performance of the proposed method. The benchmarking methods including: 1) the original method, where no control is applied for the PV inverters. The status of SCs are set to “off”, and the position of the OLTC is kept to 1.0 p.u.; 2) the MASAC-S method, where the PV inverters in the lower-level are controlled by MASAC based on local information, while the upper-level devices are scheduled to minimize voltage deviation and long-term switching numbers by SAC method according to global observation. The upper-level agent is first trained, after which the scheduling solutions made by the upper-level agent are used by the lower-level agents during the training process. This method imitates the two-level control strategy proposed in [21]. Since the objective of the upper-level agent is to minimize the long-term cost, the upper-level control method in [21] is replaced by SAC in this study for a fair comparison. It is also our effort to model each sub-network instead of each PV inverter as an agent to deal with situations when there is a high penetration level of PV; 3) the two-stage stochastic programming (SP) method, where OLTC and CBs are scheduled in the first stage, and PV inverters act in the second stage to supplement the decisions made in the first stage. The two-stage stochastic programming problem is transferred to a deterministic optimization problem and scenario reduction method is applied to obtain 40 sets of representative scenarios [17]; 4) the MASAC-C method, where PV inverters are controlled by the MASAC algorithm, and the OLTCs and SCs are controlled by SAC method.
according to global observation. The control model and the training mechanism of MASAC-C method are the same as the proposed method except that the Z-bus method [37] is utilized to calculate reward during the training process instead of the surrogate model; 5) **Centralized-C method**, where both the upper-level and lower-level devices are scheduled by an agent-based on global information. The training mechanism of Centralized-C is the same as MASAC-C method. The “Para. dep.” represents whether the corresponding method depends on the accurate physical model of ADN. It is worth noting that the MASAC-S, MASAC-C method, and Centralized-C method use Z-bus method for the calculation of reward. Therefore, accurate knowledge of the line parameters and topology is needed, see “Para. dep.” with ✓. Since the reward of the proposed method is calculated by the surrogate model, which is trained in a supervised fashion utilizing the historically recorded data, it is physical-model-free and its “Para. dep.” is ✗.

The results obtained by different methods are listed in Table IV. It can be observed that when no control is used for the scheduling of OLTCs, SCs, and PV inverters, the average voltage deviation is high. In addition, the maximum voltage deviation crosses the allowable range. When MASAC-S method is utilized, the voltage profiles are adjusted within the allowable ranges. However, since the two-level agents are trained separately in the MASAC-S method, there is a lack of systematical coordination between the two-timescale control devices. Therefore, its average voltage deviation is higher than other control strategies. The SP can achieve coordinated scheduling of OLTC, SCs, and PV inverters, and thus obtains better control performance than MASAC-S method. Since the proposed method takes decisions based on the latest observations instead of the generated scenarios, it achieves better voltage control performance than the SP method. In addition, the proposed method has similar performance as that obtained by MASAC-C method, which relies on the accurate physical model of ADN. This demonstrates that the surrogate model can provide an accurate reward signal for the training of the DRL agents. Although the PV inverters are scheduled based on local information for the proposed and MASAC-C methods, they achieve performance that is close to the Centralized-C method that takes decisions using global information. This demonstrates that the network-partition and the proposed central training and decentralized execution framework can enhance the coordination between PV inverters even based on only local information. Note that the Centralized-C method requires an accurate physical model and perfect communication links that are difficult to obtain in practice.

The averaged cumulative switching numbers obtained by various methods are also listed in Table IV. Due to the lack of inter-level coordination, the upper-level agent of MASAC-S method chooses to change the positions of OLTCs and SCs more frequently to reduce the voltage deviation. The SP method also tends to schedule the mechanical devices more frequently to adjust the voltage. By contrast, the centralized training and decentralized execution mechanism helps better exploit the capability of PV inverters and the concurrent training process with information exchange improves the systematical coordination between two kinds of assets for the proposed and the MASAC-C method. As a result, they achieve better control performance with fewer operation times.

A test day is selected to further evaluate the performance of the proposed method. The PV generation and load profiles during the test day are shown in Fig. 5 and Fig. 6, respectively. The voltage profiles obtained by various methods when \( t = 11:00 \) am are shown in Fig. 7. It can be observed that nodes 16-18 cross the upper limit when no control strategy is utilized. The disadvantages of lack of systematical coordination between two-level devices for the MASAC-S method are observed here. Since the concurrent training process with information exchange enhances the systematical coordination between two kinds of assets, the proposed and MASAC-C methods achieve better control performance than MASAC-S method. The centralized training and decentralized execution framework helps coordinate PV inverters based on local information, enabling them to achieve performances that are close to that of the Centralized-C method.
D. Evaluation of the Generalization Ability of the Proposed Method

1) Test Under Rapid Variations of PV Generations

To verify the effectiveness of the proposed two-timescale control method in terms of having fast response speed, comparative tests are carried out under situations when rapid fluctuations of PV generation occur owing to the cloud dynamics. The fast-varying PV generation profiles in 1 minute are shown in Fig. 9. Since PV generation variations happen in 1 minute while the traditional devices are typically scheduled on an hourly basis, the positions of OLTCs and statuses of SCs are kept the same during this process. The decision cycles for the MASAC-S, proposed, and the MASAC-C methods are set to 1s. This can be achieved in practice since the scheduleings of PV inverters by those methods are based on local information. The SP method employs a pre-determined solution to deal with the fast-varying PV generation. The time delay is considered for the Centralized-C method, the value of which is set to 8s, including the double-way communication time. The voltage profiles achieved by the Centralized-C method without considering the time delay are taken as the benchmark.

The voltage profiles obtained by various methods are shown in Fig. 10. It can be found that the voltage crosses the upper limit due to the inverse power flow caused by the high PV injections. The Centralized-C method fails to adjust the voltages to allowed ranges owing to the communication time delay. The negative impact of time delay on the control performance of the centralized method is observed here. Although the SP method can adjust the voltages to allowed ranges, it suffers from large voltage fluctuations since it employs a constant solution to deal with the rapid variation of PV generation during the whole process. The MASAC-S, the proposed and MASAC-C method learn a coordinated control strategy during the centralized training stage. Therefore, although only local information is utilized, these three methods can achieve coordinated scheduling of PV inverters. The proposed and MASAC-C methods further enhance the control performance through the systematical coordination between different assets learned during the concurrent training process. The performance of the proposed method is very close to that obtained by the perfect physical-model-based MASAC-C method. The benchmark method based on global information achieves the best control performance. However, it is impossible to obtain in practice owing to the time delay in the network control system.
E. Results on IEEE 123-bus System

TABLE V Comparison results for different strategies in the IEEE 123-bus system

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>1.34%</td>
<td>-</td>
<td>7.83%</td>
<td>-</td>
</tr>
<tr>
<td>MASAC-S</td>
<td>0.18%</td>
<td>40.7</td>
<td>2.0%</td>
<td>✓</td>
</tr>
<tr>
<td>SP</td>
<td>0.19%</td>
<td>41.2</td>
<td>2.36%</td>
<td>✓</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.16%</td>
<td>6.7</td>
<td>1.96%</td>
<td>x</td>
</tr>
<tr>
<td>MASAC-C</td>
<td>0.15%</td>
<td>7.2</td>
<td>2.05%</td>
<td>✓</td>
</tr>
<tr>
<td>Centralized-C</td>
<td>0.13%</td>
<td>7.4</td>
<td>1.89%</td>
<td>✓</td>
</tr>
</tbody>
</table>

Fig. 12. Voltage profiles achieved by various methods in the test when $t=12:00$ am.

The comparison results for different control strategies on the IEEE 123-bus system are shown in Table V. It can be found that the maximum voltage deviation crosses the allowable range. The MASAC-S can adjust the voltages to allowed ranges. However, due to the lack of coordination between the two kinds of assets, it chooses to change the status of mechanical devices more frequently to reduce the voltage deviations. The SP method also tends to schedule OLTC and SCs more frequently for voltage regulation. Since the information exchange and concurrent training process enhance the inter-level coordination of the proposed method, it achieves better control performance than MASAC-S method with fewer switching numbers of mechanical devices. It also obtains a similar performance like that by MASAC-C method. The latter relies on an accurate physical model, illustrating the effectiveness of the proposed surrogate-model-enabled physical-model-free method. As the PV inverters are scheduled based on global information, the Centralized-C method has the best control performance. However, it requires perfect communication links and is fragile to a single-point failure. The network partition and centralized training and decentralized execution framework enhance the coordination between PV inverters and help the proposed method achieve a performance that is close to the Centralized-C method.

A sunny day is selected to further evaluate the performance of the proposed method. The voltage profiles obtained by different methods when $t=12:00$ are shown in Fig. 12. It can be observed from the figure that when no control strategy is applied, the voltages of nodes 65-67, 70-123 cross the upper limit. The proposed method can adjust the voltages to allowed ranges without the need for an accurate system model. The results are consistent with that we have observed in Table V.

The voltage prediction errors of the proposed surrogate model on this test day are shown in Fig. 13. The errors are less than 1.0e-3 p.u. in the morning and evening. Due to the high PV generations around noon, the prediction errors become larger in this period. However, the largest error is still less than 5.0e-3 p.u. which is relatively small, demonstrating the high accuracy of the proposed surrogate model.

Fig. 13. Voltage prediction errors of the surrogate model on the test day.

To evaluate the feasibility of the proposed method under extreme situations, further tests are carried out when there are large deviations between the actual load demands and the forecasted values. Four cases are considered: 1) Case 1: all the random coefficients of load demand are within the range [0.8, 1.2]; 2) Case 2: 20 randomly selected nodes to deviate 30% from the forecasted ones; 3) Case 3: 30 randomly selected nodes to deviate 30% from the forecasted ones; 4) Case 4: 40 randomly selected nodes to deviate 30% from the forecasted ones. The voltage profiles achieved by the proposed method on the test day on the IEEE 123-bus system are shown in Fig. 14. It can be observed that the voltage drop becomes larger when there are more nodes with large deviations, see Case 4 for example. This is because the extreme cases are different from situations during the training of the agent. However, both the maximum rise and drop of voltage profiles by the proposed method are less than 0.02 p.u., demonstrating that the proposed method can provide feasible solutions even under extreme situations.

F. Results on IEEE 342-node System

TABLE VI Comparison results for different strategies on the 342-node system

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>2.53%</td>
<td>-</td>
<td>6.88%</td>
<td>-</td>
</tr>
<tr>
<td>MASAC-S</td>
<td>1.48%</td>
<td>6.7</td>
<td>4.99%</td>
<td>✓</td>
</tr>
<tr>
<td>Proposed</td>
<td>1.45%</td>
<td>7.2</td>
<td>4.99%</td>
<td>x</td>
</tr>
<tr>
<td>Centralized-C</td>
<td>1.43%</td>
<td>4.7</td>
<td>4.98%</td>
<td>✓</td>
</tr>
</tbody>
</table>

To further evaluate the scalability of the proposed method, comparative tests are carried out on IEEE 342-node low voltage networked test system, which is a representative of a moderate size urban system [34]. The comparison results for different strategies on the test system are shown in Table VI. It can be observed from the table that when no control is applied, the maximum voltage deviation crosses the allowable range. The
MASAC-S method can adjust the voltage to allowable ranges. However, it require the accurate physical model of the ADN. The proposed physical-model-free method can achieve better control performance without the requirement of accurate knowledge of physical model. The network partition and centralized training and decentralized execution framework enable the proposed method to achieve performance that is close to that obtained by the Centralized-C method which requires the global information and accurate system model. Voltage profiles achieved by various methods on a sunny day in the test set is plotted in Fig. 15. That the proposed method can adjust the voltages to allowed ranges as well as mitigate voltage deviations, demonstrating the effectiveness of the proposed method.

![Fig. 15. Voltage profiles achieved by various methods on a sunny day in the test set.](image)

V. CONCLUSION

This paper proposes a physical-model-free two-timescale control framework for the voltage regulation of ADN utilizing OLTCs, SCs, and PV inverters. The proposed method can achieve coordinated control of two-timescale devices and provide fast decisions to reduce the voltage variations caused by the rapid changes of PV generations. It also mitigates the dependence on an accurate physical system model by developing a novel interaction mechanism between the surrogate model and DRL agent. Comparative tests with several benchmark methods demonstrate that: 1) the proposed GP-based surrogate model can accurately estimate the voltage magnitude given the power injection of each node; 2) the proposed surrogate-model-enabled physical-model-free method can obtain similar control performance as that achieved by the method based on the accurate physical model; 3) the network-partition and centralized training and decentralized execution framework enhance the coordination between PV inverters and help the proposed method achieve performance close to centralized method with global information; 4) the proposed method can effectively deal with the fast voltage fluctuations caused by the rapid variations of PV generations; 5) the concurrent training process and information exchange help achieve systematic coordination of the two-timescale devices.
Di Cao is currently working toward the Ph.D. degree in control science and engineering at the University of Electronic Science and Technology of China. His research interest includes optimization of distribution network and applications of machine learning in power systems.

Junbo Zhao (SM’19) has been an Assistant Professor at Mississippi State University, Starkville, MS, USA since 2019. He received the Ph.D. degree from the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA, in 2018. He was a Research Assistant Professor at Virginia Tech from May 2018 to August 2019. He did the summer internship at Pacific Northwest National Laboratory from May to August 2017. He is currently the chair of the IEEE Task Force on Power System Dynamic State and Parameter Estimation and the IEEE Task Force on Cyber-Physical Interdependency for Power System Operation and Control, co-chair of the IEEE Working Group on Power System Static and Dynamic State Estimation, the Secretary of the IEEE PES Bulk Power System Operation Subcommittee.

He has published three book chapters and more than 100 peer-reviewed journal and conference papers, where more than 60 appear in IEEE Transactions. His research interests are cyber-physical power system modeling, estimation, security, dynamics and stability, uncertainty quantification, renewable energy integration and control, robust statistical signal processing and machine learning. He serves as the editor of IEEE Transactions on Power Systems, IEEE Transactions on Smart Grid and IEEE Power and Engineering Letters, the Associate Editor of International Journal of Electrical Power & Energy Systems, and the subject editor of IET Generation, Transmission & Distribution. He is the recipient of best paper awards of IEEE PES General Meeting at 2020 and 2021, and 2019 IEEE PES ISGT Asia. He received the Top 3 Associate Editor Award from IEEE Transactions on Smart Grid and IEEE PES Outstanding Engineering Award in 2020. He has been listed as 2020 World’s Top 2% Scientists released by Stanford University.

Weihao Hu (S’06–M’13–SM’15) received the B.Eng. and M.Sc. degrees from Xi’an Jiaotong University, Xi’an, China, in 2004 and 2007, respectively, both in electrical engineering, and Ph. D. degree from Aalborg University, Denmark, in 2012. He is currently a Full Professor and the Director of Institute of Smart Power and Energy Systems (ISPES) at the University of Electronics Science and Technology of China (UESTC). He was an Associate Professor at the Department of Energy Technology, Aalborg University, Denmark and the Vice Program Leader of Wind Power System Research Program at the same department. His research interests include artificial intelligence in modern power systems and renewable power generation. He has led/participated in more than 15 national and international research projects and he has more than 200 publications in his technical field.

He is an Associate Editor for IET Renewable Power Generation, a Guest Editor-in-Chief for Journal of Modern Power Systems and Clean Energy Special Issue on Applications of Artificial Intelligence in Modern Power Systems, a Guest Editor-in-Chief for Transactions of China Electrical Technology Special Issue on Planning and operation of multiple renewable energy complementary power generation systems, and a Guest Editor for the IEEE TRANSACTIONS ON POWER SYSTEMS Special Section on Enabling very high penetration renewable energy integration into future power systems. He was serving as the Technical Program Chair (TPC) for IEEE Innovative Smart Grid Technologies (ISGT) Asia 2019 and is serving as the Conference Chair for the Asia Energy and Electrical Engineering Symposium (AEEES 2020). He is currently serving as Chair for IEEE Chengdu Section PELS Chapter. He is a Fellow of the Institution of Engineering and Technology, London, U.K. and an IEEE Senior Member.

Nanpeng Yu (Senior Member, IEEE) received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2006, and the M.S. and Ph.D. degrees in electrical engineering from Iowa State University, Ames, IA, USA, in 2007 and 2010, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA, USA. His current research interests include machine learning in smart grid, electricity market design and optimization, and smart energy communities. He is an Editor of IEEE Transactions on Smart Grid and IEEE Transactions on Sustainable Energy.

Fei Ding (Senior Member, IEEE) received the Ph.D. degree from Case Western Reserve University, and she joined the National Renewable Energy Laboratory as a Research Engineer in 2015. Her research focuses on distribution system automation and optimization, distribution system modeling and simulation, renewable energy grid integration, advanced distribution management system, and smart grid resilience.
Qi Huang (Senior Member, IEEE) was born in Guizhou province in the People’s Republic of China. He received the B.S. degree in electrical engineering from Fuzhou University, Fuzhou, China, in 1996, the M.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree in electrical engineering from Arizona State University, Tempe, AZ, USA, in 2003. He is currently a Professor with Chengdu University of Technology (CDUT) and University of Electronic Science and Technology of China (UESTC). He is vice president of CDUT and the Director with the Sichuan State Provincial Lab of Power System Wide-area Measurement and Control. He is an IET Fellow and IEEE Senior Member. His current research and academic interests include power system instrumentation, power system monitoring and control, and informatics for smart electric energy systems.

Zhe Chen (M’95–SM’98–F’19) received the B.Eng. and M.Sc. degrees from Northeast China Institute of Electric Power Engineering, Jilin, China, and the Ph.D. degree from University of Durham, Durham, U.K. He is a Full Professor with the Department of Energy Technology, Aalborg University, Denmark. He is the Leader of Wind Power System Research Program in the Department of Energy Technology, Aalborg University and the Danish Principle Investigator for Wind Energy of Sino-Danish Centre for Education and Research. His research areas include power systems, power electronics and electric machines; and his main current research interests are wind energy and modern power systems. He has led many research projects and has more than 400 publications in his technical field.

Dr. Chen is an Editor of the IEEE TRANSACTIONS ON POWER SYSTEMS, an Associate Editor of the IEEE TRANSACTIONS ON POWER ELECTRONICS, a Fellow of the Institution of Engineering and Technology, London, U.K., a Chartered Engineer in the U.K., and a Fellow of the IEEE.