Automatic Voltage Control of Differential Power Grids Based on Transfer Learning and Deep Reinforcement Learning

Tianjing Wang, Member, IEEE, and Yong Tang, Senior Member, IEEE, Fellow, CSEE

Abstract—In terms of model-free voltage control methods, when the device or topology of the system changes, the model’s accuracy often decreases, so an adaptive model is needed to coordinate the changes of input. To overcome the defects of a model-free control method, this paper proposes an automatic voltage control (AVC) method for differential power grids based on transfer learning and deep reinforcement learning. First, when constructing the Markov game of AVC, both the magnitude and number of voltage deviations are taken into account in the reward. Then, an AVC method based on constrained multi-agent deep reinforcement learning (DRL) is developed. To further improve learning efficiency, domain knowledge is used to reduce action space. Next, distribution adaptation transfer learning is introduced for the AVC transfer circumstance of systems with the same structure but distinct topological relations/parameters, which can perform well without any further training even if the structure changes. Moreover, for the AVC transfer circumstance of various power grids, parameter-based transfer learning is created, which enhances the target system’s training speed and effect. Finally, the method’s efficacy is tested using two IEEE systems and two real-world power grids.

Index Terms—Deep reinforcement learning, differential power grids, transfer, voltage control.

NOMENCLATURE

A. Variables

\( s, a, o \) State, action, observation.

\( M \) Markov game.

\( r \) Reward.

\( \pi \) Policy.

\( \mathcal{P} \) Transition probability.

\( V \) Node voltages.

\( VR \) Voltage regulators.

\( TC \) On-load tap changers.

\( CR \) Switchable capacitors/reactors.

\( V_{\text{ref}}, V_{\text{ref}}, V_{\text{ref}} \) Reference voltage.

\( N_B, N_{VB}, N_{DB} \) Number of all nodes, violation nodes and divergent nodes.

\( P_{\text{loss}} \) Active power loss.

\( \kappa_i \) Action amount of device \( i \).

\( N_c \) Action number of control devices.

\( \gamma_t \) Discount factor at time \( t \).

\( U(s) \) State-value function.

\( Q(s, a) \) Action value function.

\( \mu \) Deterministic parametric policy.

\( J \) Policy performance.

\( \theta \) Policy parameter.

\( D \) Experience buffer.

\( E_{x,a} \sim D \) Expected return.

\( \mathcal{L} \) Cost function.

\( H \) Entropy.

\( D \) Memory buffer.

\( \Omega_{\text{NV}} \) Abnormal voltage node set.

\( V_i \) Abnormal voltage \( i \).

\( N_{\text{NV}} \) Number of abnormal voltage.

\( \delta_{QR}^i \) Nearest reactive power regulation devices of the \( i \)th abnormal voltage node.

\( \Sigma_{QR} \) Reactive power regulation device set.

\( \mathcal{D} \) Domain.

\( T \) Task.

\( p(x) \) The probability distribution \( p(x) \) of the population \( x \).

\( \mathcal{X} \) Feature space.

\( \mathcal{Y} \) Label space.

\( D_{\text{dis}} \) Distance.

\( \sigma \) Balance factor.

\( c \) Class label.

\( \mathcal{H} \) Reproducing kernel Hilbert space.

\( \mathcal{X} \) Input data matrix.

\( A, I, H \) Transformation matrix, identity matrix and centering matrix.

\( M_0, M_c \) MMD matrices.

\( \xi_S \) Source decision list.

\( \xi_T \) Target decision list.

\( h \) Homomorphism.

\( f, g \) Mapping function.

B. Parameters

\( S, A, O \) State, action and observation space.

\( \phi \) Parameter of approximation.

\( \lambda \) Weight of entropy.
I. INTRODUCTION

With continuous penetration of renewable energy resources, rapid response and control of voltage becomes more and more important to maintain system stability [1], [2]. Autonomous voltage control (AVC) methods are developed to manage system voltage levels and reactive power flows, which can control all the voltage regulating devices to maintain voltage magnitudes within the desirable range [3]. Existing AVC research mainly focuses on model-based control methods. This method can be divided into three directions, i.e., transforming the AVC problem into a deterministic optimization problem [4]–[6], adding uncertainty of distributed energy resources into the AVC problem [7], [8], and decentralized control methods [9], [10]. However, various random power supplies and loads of modern power systems make the model-based methods very challenging to realize large-scale real-time voltage control. Besides, acquiring an accurate model of some non-linear system components is incredibly difficult [11].

Driven by recent advances in artificial intelligence (AI), some scholars have primarily explored some AI-based voltage control methods. First, J. G. Vlachogiannis et al. proposed a reactive power control method based on reinforcement learning [12]. However, the Q table of reinforcement learning uses little data, and learned policies are not generalized enough. By combining reinforcement learning with deep learning, the researchers in [13] proposed an AVC method using deep Q-network (DQN) and deep deterministic policy gradient (DDPG) to fit the policy of Q learning better. Considering the size of the system and the number of control elements, [11] developed a multi-agent AVC algorithm based on the multi-agent DDPG (MADDPG) method to solve the AVC problem. Moreover, [14] developed a safe off-policy DRL algorithm to keep voltage from fluctuating during the control process for the AVC control problem. [15] developed a voltage regulation strategy for distribution grids on two-time scales by combining data-driven with physics-based optimization, taking into account the response period of voltage control.

While the current AI-based AVC approaches have achieved good performance, there are still some shortcomings and challenges, e.g., poor circumstance transferability [16], long-time data training, and waste of domain knowledge [17].

For the challenges of poor circumstance transferability and long-time data training, this study proposes an AVC method based on transfer learning and deep reinforcement learning to transfer some parameters or data from the source domain to the target domain. At present, several scholars conducted early research into the application of transfer learning to power systems. Some researchers transfer the knowledge of the Q value table directly. [18] offered a transition method for speeding computing from already gained information and built an information transfer methodology between power flow snapshots. A behavior transfer was used in consensual transfer reinforcement learning in [19] to leverage past knowledge of original tasks for a new optimization goal based on their similarities. Others alter the desired data distribution. By repeatedly decreasing the marginal and conditional distribution discrepancies between the trained and new data, [20] transferred a learned dynamic security assessment to a distinct but interrelated fault. Transfer learning can improve the ability of circumstance transfer and shorten training time. However, current transfer learning applications are limited and simplistic, with no systematic consideration of diverse transfer circumstances. In this paper, transfer learning applications are split into the transfer circumstance of systems with the same structure but distinct topological relations/parameters and the transfer circumstance of various power grids, based on the variation extent of transfer targets. Different from the original power grid, these two types of power grids are uniformly defined as differential power grids.

Additionally, domain knowledge is introduced to reduce the action space of DRL. This study utilizes the multi-agent DRL because the voltage control problem is a local problem, and the interaction between node voltages can be balanced through the game. The general flow chart of the proposed method is shown in Fig. 1. The upper part of the figure is the control process of AVC, and the lower part is the transfer learning methods under various transfer circumstances. Thus, the main contributions of this study are as follows:

1) The Markov game of AVC is established by defining state, action, policy, and reward. The reward not only considers the magnitude of voltage deviation, but also includes the number of voltage deviation nodes.

2) An AVC method is proposed based on constrained multi-agent DRL and voltage control strategy. To cope with the limitation of voltage fluctuation in the control process, a penalty constraint is added to the action of DRL. In addition, the control range is constrained by locating reactive power compensation devices near abnormal voltage nodes to improve learning efficiency.

3) Based on distribution adaption, a transfer learning approach is provided for AVC transfer circumstance of systems with the same structure but distinct topological

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$</td>
<td>Numbers of sub policies.</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Regulation parameter.</td>
</tr>
<tr>
<td>$\alpha_{r1}, \alpha_{r2}$</td>
<td>Reward coefficient.</td>
</tr>
<tr>
<td>$C_D$</td>
<td>Electricity price.</td>
</tr>
<tr>
<td>$C_T$</td>
<td>Action cost of device $i$.</td>
</tr>
<tr>
<td>$T$</td>
<td>Time horizon.</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of agents.</td>
</tr>
</tbody>
</table>

C. Abbreviations

AVC: Automatic voltage control.
DRL: Deep reinforcement learning.
DQN: Deep Q-network.
DDPG: Deep deterministic policy gradient.
MADDPG: Multi-agent DDPG.
MDP: Markov decision-making process.
BDA: Balanced distribution adaptation.
MMD: Maximum mean discrepancy.
CSR: Constraint satisfying ratio.
PPO: Proximal policy optimization.
QMIX: Q learning with mixed networks.
MAPPO: Multi-agent PPO.
SAC: Soft actor-critic.

W
relations/parameters. Through transfer learning, data of the target domain can be mapped to the same distribution as the source domain by a transformation matrix. The target samples can achieve good results on the source model without further training.

4) Parameter-based transfer learning is built for the AVC transfer situation of various power grids. For parameter transfer, in addition to parameter transfer of neural networks, it also transfers the policy of DRL and takes the policy learned in the source domain as an option for the target domain, thus improving training speed and effect of the target system.

The remainder of this study is as follows. The Markov game of AVC is introduced in Section II. Section III presents an AVC method based on constrained multi-agent DRL and voltage control strategy. Section IV provides AVC approaches for differential power grids with transfer learning. Experiment findings in Section V indicate the efficacy of the suggested technique. Section VI concludes with some final observations.

II. MARKOV GAME OF AVC

The Markov decision-making process (MDP) is a model of sequenced decision-making that can be leveraged to emulate random rules intelligent agents can implement in a Markovian system state [21]. Voltage control is likewise a decision process of regulation inspired by real-time feedback, hence MDP may be used to depict it. Considering the game relationship between different agents, the MDP for multi-agents is called the Markov game.

The main elements in the Markov game include state $s \in S$, action $a_i \in A_i$, observation $o_i \in O_i$, policy $\pi_i: O_i \times A_i \rightarrow [0, 1]$ and reward $r_i: S \times A \rightarrow \mathbb{R}$ for agent $i$. Where $\pi_i$ is a mapping from observation to action and is a transition probability $P_T$ of action, and $r_i$ is a function of the state and the joint action. The AVC’s associated elements are defined as follows:

- State, action and observation

For the AVC, the state is system-wide bus voltage magnitudes, as shown in $s = [V]$. The action targets are reactive power regulating devices that can be controlled, including voltage regulators, on-load tap changers, and switchable capacitors/reactors. Therefore, the action space can be constructed as $a = [VR, TC, CR]$. A regional measure of bus voltage magnitudes is denoted as the observation of agent $i$, that is $o_i = [V_i]$. Since the action of MADDPG is continuous, the action space in the algorithm setting is $[0, 3]$, where $[0, 1]$, $[1, 2]$, and $[2, 3]$ correspond to $VR$, $TC$, and $CR$, respectively. $VR$ and $TC$ are continuous variables, which can directly map actions to control these two variables. Since $CR$ is discrete, $[2, 3]$ should be divided equally according to the number of $CR$, and each interval after equal division corresponds to the action of a different $CR$.

- Reward

For DRL, the reward is set to evaluate the rationality of action, which is closely related to learning efficiency. After each action of DRL, the state of $VR/TC/CR$ changes, which changes the reactive power distribution and node voltages of the system. The reward shall reflect the current state of system voltage and rationality of action. Referring to [13], the reward settings of node voltage $i$ in different states are shown in Table I. Where $\alpha_{r1}$ is the reward coefficient. For most situations, the reward should be set between $-1$ and $1$ to promise similar sensitivity of the neural network relative to the reward, and it can be slightly larger when it is incredibly infeasible. Thus, $\alpha_{r1}$ is set as 3 in this paper. $V_{\text{low}}^\text{ref}$, $V_{\text{ref}}$ and $V_{\text{up}}^\text{ref}$ are set according to different voltage control objects.

<table>
<thead>
<tr>
<th>$V_{\text{ref}}$</th>
<th>$V_{\text{up}}^\text{ref}$</th>
<th>$r_1$’s monotone when $V_i \rightarrow V_{\text{ref}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[V_{\text{ref}}, V_{\text{up}}^\text{ref}]$</td>
<td>$V_{\text{ref}} - V_{\text{low}}^\text{ref}$</td>
<td>$0 \rightarrow 1$</td>
</tr>
<tr>
<td>$[V_{\text{low}}^\text{ref}, V_{\text{ref}}]$</td>
<td>$V_{\text{up}}^\text{ref} - V_{\text{ref}}$</td>
<td>$0 \rightarrow 1$</td>
</tr>
<tr>
<td>$[V_{\text{up}}^\text{ref}, 1.25]$</td>
<td>$V_{\text{ref}} - 1.25$</td>
<td>$-1 \rightarrow -0.2$</td>
</tr>
<tr>
<td>$[0.8, V_{\text{low}}^\text{ref}]$</td>
<td>$0.8 - V_{\text{ref}}$</td>
<td>$-1 \rightarrow -0.25$</td>
</tr>
<tr>
<td>$[1.25, \infty]$</td>
<td>$-\alpha_{r1}$</td>
<td>No change</td>
</tr>
<tr>
<td>$[0, 0.8]$</td>
<td>$-\alpha_{r1}$</td>
<td>No change</td>
</tr>
</tbody>
</table>

Voltage deviation is generally the voltage decrease or increase in a particular area. DRL-based AVC methods only consider voltage deviation magnitude and do not consider the...
number of voltage deviation nodes. The number of voltage deviation nodes can also reflect the voltage condition of the system. Therefore, when setting the reward, the voltage deviation magnitude and number of voltage deviation nodes are considered simultaneously.

When voltage is in the normal range of \([V_{\text{low}}^{\text{ref}}, V_{\text{ref}}^{\text{up}}]\), the reward is the average value of all nodes, which is a positive value. When voltage is in the range of \([0.8, V_{\text{low}}^{\text{ref}}]\) or \([V_{\text{ref}}^{\text{up}}, 1.25]\), that is to say, the node has small voltage deviation; the reward is the average value of the reward of all swing nodes, which is a slightly negative value. When voltage is at \([0, 0.8]\) or \([1.25, \infty]\), it belongs to the divergence range, and the reward is a maximum negative value. Therefore, the reward \(r_V\) is set as follows:

\[
\begin{align*}
    r_V = \begin{cases} 
        \sum_{i=1}^{N_B} \frac{r_i}{N_B} & \text{if } V_i \in \left[ V_{\text{low}}^{\text{ref}}, V_{\text{ref}}^{\text{up}} \right] \\
        \sum_{i=1}^{N_{\text{VB}}} \frac{r_i}{N_{\text{VB}}} & \text{if } V_i \in \left[ 0, 0.8 \right] \cup \left[ 1.25, \infty \right] \\
        -\alpha_{r_1} & \text{if } V_i \in \left[ 0, 0.8 \right] \cup \left[ 1.25, \infty \right]
    \end{cases} 
\end{align*}
\]

where \(N_B\) and \(N_{\text{VB}}\) refer to the number of all nodes and violation nodes.

Considering the number of voltage deviation nodes, when voltage is in the range of \([V_{\text{low}}^{\text{ref}}, V_{\text{ref}}^{\text{up}}]\), the reward is a fixed positive value. When voltage is at \([0.8, V_{\text{ref}}^{\text{up}}]\) or \([V_{\text{ref}}^{\text{up}}, 1.25]\), the reward is proportional to the number of swing nodes. When voltage is at \([0, 0.8]\) or \([1.25, \infty]\), it is also proportional to the number of divergent nodes. Then the reward \(r_N\) is:

\[
\begin{align*}
    r_N = \begin{cases} 
        1 & \text{if } V_i \in \left[ V_{\text{low}}^{\text{ref}}, V_{\text{ref}}^{\text{up}} \right] \\
        (-\alpha_{r_1}) \frac{N_{\text{DB}}}{N_B} & \text{if } 3V_i \in \left[ 0, 0.8 \right] \cup \left[ 1.25, \infty \right] \\
        (-\alpha_{r_1}) \frac{N_{\text{DB}}}{N_B} & \text{if } 3V_i \in \left[ 0, 0.8 \right] \cup \left[ 1.25, \infty \right]
    \end{cases} 
\end{align*}
\]

where \(N_{\text{DB}}\) denotes the number of divergent nodes.

In addition, to make the control result and control cost reasonable, the influence of AVC on system operation cost and control cost should be considered [14], and the reward \(r_C\) is set as follows.

\[
r_C = -\alpha_{r_2} \left[ C_p P_{\text{loss}} + \sum_{i=1}^{N_c} C_i^T \kappa_i \right]
\]

where \(C_p\) and \(C_i^T\) are electricity price and action cost of device \(i\), respectively; \(P_{\text{loss}}\) is the active power loss; \(\kappa_i\) is the action amount of device \(i\); \(N_c\) is current action number of control devices; \(\alpha_{r_2}\) is the reward coefficient of control cost. In this study, \(\alpha_{r_2}\) is fixed to 0.1 to keep the reward in the same scale as previous rewards.

To sum up, the total reward \(r\) is as shown below:

\[
r = r_V + r_N + r_C
\]

III. AVC BASED ON CONSTRAINED MULTI-AGENT DRL AND VOLTAGE CONTROL STRATEGY

A. Constrained Multi-agent DRL

The DRL’s purpose is to discover a policy that maximizes the anticipated discounted return:

\[
\max_{\pi} \pi \in \mathcal{P} \left[ \sum_{t=0}^{T} \gamma_t r_t \right]
\]

where \(T\) is the time length; \(\gamma_t\) is the discounted coefficient at time \(t\).

State-value function \(U(s)\) and action-value function \(Q(s, a)\) can be defined below:

\[
U(s) = \mathbb{E}_{a_t \sim \pi, s_{t+1} \sim \mathcal{P}_r(s_t, a_t)} \left[ \sum_{t=0}^{T} \gamma_t r_t | s_0 = s \right] \\
Q(s, a) = \mathbb{E}_{a_t \sim \pi, s_{t+1} \sim \mathcal{P}_r(s_t, a_t)} \left[ \sum_{t=0}^{T} \gamma_t r_t | s_0 = s, a_0 = a \right]
\]

To address the AVC issue, which is described as a Markov game in Section II, we develop a DRL technique called constrained multi-agent DRL in this section. Among the reinforcement learning algorithms in a multi-agent environment, several common algorithms include MADDPG, MAPPO, QMIX, and so on. Compared with other algorithms, MADDPG is more suitable for training multi agents because of its higher sampling efficiency in the case of limited computing resources. Furthermore, the algorithm can be used not only in a cooperative environment but also in a competitive environment. Considering the limitations of voltage fluctuation and control range in the control process, the off-limit of MADDPG action is punished, and the control range is determined to reduce the control risk and control cost.

1) Multi-Agent DRL Construction Based on MADDPG

For traditional multi-agent reinforcement learning, the environment is changing and unstable for each agent, resulting in the DQN experience not being directly used and aggravating deviation of the gradient descent algorithm. MADDPG [22] uses centralized training and distributed execution approach to tackle the issue. During training, the Critic and Actor are given all of the agent’s information at a centralized point. The Actor can only execute with its local information throughout execution, which keeps the environment consistent. The following is the model’s design:

- Actor and Critic design of multi-agents

The MADDPG uses the deterministic parametric policy \(\mu_i : S_i \rightarrow A_i\), approximated by a neural network named Actor for agent \(i\). For start state of the episode, the measure of policy performance \(J(\theta_1^a)\) for agent \(i\) is the state-value function:

\[
J(\theta_1^a) = U_i(s^0)
\]

Gradient ascent is utilized to update the Actor, causing it to obey the policy in the orientation of (8)’s gradient. Each agent develops a Critic who requires records and an Actor who requires local storage. As a result, the gradient formula \(\nabla_{\theta_i} J\) with regard to agent \(i\)’s policy parameter \(\theta_i\) is:

\[
\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{s, a_0 \sim D} \left[ \nabla_{\theta_i} \mu_i(a_i|o_i) \nabla_{a_i} Q_i^a(s, a_1, \ldots, a_n) | o_i = \mu_i(o_i) \right]
\]

where \(n\) is the number of agents; \(D\) refers to the memory buffer, whose components are \((s, a', s', r, \gamma, \cdot, \cdot)\); \(\mathbb{E}_{s, a_0 \sim D}\) is expected return; \(\mu_i(o_i|a_i)\) is continuous policy parameter of agent \(i\); \(Q_i^a(s, a_1, \ldots, a_n)\) is the centralized action-value function.
• Evaluating policies of other agents

Policies of other agents are evaluated to reduce the need for constant communication between agents. Agent \(i\) approximates the policy \(\hat{\mu}_{\phi_i}^{(k)}\) of agent \(j\) (where \(\phi\) is the approximated parameter). The cost \(L\) of each agent with the policy’s entropy is as follows.

\[
L(\phi_i^j) = -E_{\alpha, \theta_i} [\log \hat{\mu}_{\phi_i}^{(k)}(a_j|\theta_i)] + \lambda H(\hat{\mu}_{\phi_i}^{(k)})
\]

where \(H(\hat{\mu}_{\phi_i}^{(k)})\) is the entropy of the policy; \(\lambda\) is the weight of entropy.

• Optimization of policy sets

A policy set is leveraged to tackle the problem of dynamic unsteadiness in a multi-agent environment. Agent \(i\)’s policy is made up of \(K\) sub-policies, and each episode is trained with one of them \(\mu_{\phi_i}^{(k)}\) (abbreviated \(\mu_i^{(k)}\)). Maximize the reward of the policy set for each agent:

\[
J_{\theta_i}(\mu_i) = E_{k \sim \text{uni}(1,K), \alpha \sim \mathcal{P}_i, \gamma \sim \mu_{\phi_i}^{(k)}} \left[ \sum_{t=0}^{T} \gamma^t r_{i,t} \right]
\]

Moreover, for each sub policy, a memory buffer \(D_i^{(k)}\) is created. The following is the changed gradient for each sub policy:

\[
\nabla_{\theta_i^{(k)}} J_{\theta_i}(\mu_i) = \frac{1}{K} E_{x \sim \mathcal{P}_i} D_i^{(k)} \left[ \nabla_{\theta_i^{(k)}} Q_{\theta_i}(x, a_1, \ldots, a_n)|a_i = \mu_i^{(k)}(a_i) \right]
\]

2) Constrained Multi-agent DRL

To limit unreasonable voltage fluctuation caused by actioning reactive power regulation devices in the control process, it is necessary to limit every action in the DRL [15]. According to (1), the limit of voltage fluctuation can be equivalent to \(r_V < \alpha_r\). For the state-value function of agent \(i\), the constraint can be set to \(E[\sum_{t=0}^{T} \gamma^t r_{V,i,t}] < (1 - \alpha)\alpha_r\). Then the optimal policy of constrained multi-agent DRL is as follows:

\[
\max E \left[ \sum_{t=0}^{T} \gamma^t r_{i,t} \right] \quad \text{s.t.} \quad E \left[ \sum_{t=0}^{T} \gamma^t r_{V,i,t} \right] < \frac{(1 - \gamma)\alpha_r}{1 - \gamma}
\]

The Lagrange equation of (13) is as follows:

\[
L(\lambda) = E \left[ \sum_{t=0}^{T} \gamma^t r_{i,t} \right] + \lambda \left( \frac{(1 - \gamma)\alpha_r}{1 - \gamma} - E \left[ \sum_{t=0}^{T} \gamma^t r_{V,i,t} \right] \right)
\]

The solution method and convergence proof of the constrained optimal strategy are shown in [15], which are not described in this study.

B. Voltage Control Strategy

To narrow the range of reactive power control and improve learning efficiency, this paper uses domain knowledge to locate the area of voltage deviation, to determine the target of reactive power control and uses reinforcement learning to determine the amount of reactive power control. Specific voltage control measures are as follows:

As shown in Fig. 2, first, according to voltage magnitude, abnormal voltage nodes are selected to form an abnormal voltage node-set \(\Omega_NV = \{V_{NV|i} = 1, 2, \ldots, N_{NV}\}\). According to the degree of voltage deviation from the constraint range, the nodes with abnormal voltage are sorted, and the sort \(S_{NV}\) can be obtained.

![Fig. 2. Voltage control strategy.](image)

Then, find the reactive power regulation devices nearest the abnormal voltage nodes. Therefore, the reactive power regulation devices nearest the \(S_{NV}\) are obtained:

\[
\Sigma_{QR} = \{\delta_{QR}^1, \delta_{QR}^2, \ldots, \delta_{QR}^{N_{NV}}\}
\]

where \(\delta_{QR}\) denotes the nearest reactive power regulation devices of the \(i\)th abnormal voltage node.

Finally, the action of reinforcement learning, whose range is \([0, 1]\), is mapped to the top \(a \cdot 100\%\) of \(\Sigma_{QR}\):

\[
a \rightarrow \{\delta_{QR}^{k} | k \in \Sigma_{QR}\}
\]

IV. DIFFERENTIAL POWER GRID VOLTAGE CONTROL BASED ON TRANSFER LEARNING

The control efficacy of the model-free AVC approach suggested above is typically lessened when the control object changes, such as changing system structure or selecting a different system. Retraining will result in squandering of previously acquired information. To tackle this problem, this study introduces transfer learning. Transfer learning is split into the transfer of systems with the same structure but distinct topological relations/parameters and various power grids, as illustrated in Fig. 3, based on the extent of control object change. Concrete methodology is described below.

A. Transfer Learning

For transfer learning, we need to make clear two definitions, which are “domain” and “task” [23]. The “domain” \(D\) contains two elements: the sample feature space \(X\) and the probability distribution \(p(x)\) of the population \(x\), in which the sample set \(X = (x_1, x_2, \ldots, x_n) \in X\) is total sample, \(P(X) = \prod_{i=1}^{n} p(x_i)\). The two domains are different if and only if \(X\) or \(P(X)\), at least one is different. There are also two parts in “task” \(T\): label space \(Y\) and function \(f_T\). The label is \(y\), and the set \(Y = (y_1, y_2, \ldots, y_m) \in Y\) is the whole sample, where the label space of \(y\) is called \(Y\). The training sample \(\{x_i, y_i\}\) suggests that \(f_T : X \rightarrow Y\) can be obtained, where
\(x_i \in \mathcal{X}, y_i \in \mathcal{Y}\). From the perspective of probability, \(f_T\) can be considered as conditional probability \(p(y|x)\), and the task can be expressed as \(T = \{ \mathcal{Y}, p(y|x) \}\). The two tasks are various if and only if at least one of the label space \(\mathcal{Y}\) or the conditional probability \(p(y|x)\), is different. For the problem in this paper, the sample feature \(x\) and the label \(y\) refer to \(s\) and \(r\) defined in Section II.

Transfer learning is the research of how to use source domain data and task to enhance the learning effectiveness of target domain data given source domain \(\mathcal{D}_S\) and task \(\mathcal{T}_S\), target domain \(\mathcal{D}_T\) and task \(\mathcal{T}_T\). Usually, \(\mathcal{D}_S \neq \mathcal{D}_T\) or \(\mathcal{T}_S \neq \mathcal{T}_T\).

### B. Transfer Learning of Systems with the Same Structure but Distinct Topological Relations/Parameters

For the same power grid, the change of grid structure or devices can cause change in data distribution. In this circumstance, \(\mathcal{D}_S, \mathcal{T}_S, \mathcal{D}_T\) and \(\mathcal{T}_T\) refer to the domain and control task before and after the change. Although the feature space \(\mathcal{X}_S = \mathcal{X}_T\) and label space \(\mathcal{Y}_S = \mathcal{Y}_T\), there are differences in marginal distribution or conditional distribution between the source domain and target domain (i.e., \(p(x_S) \neq p(x_T)\) or \(p(y_S|x_S) \neq p(y_T|x_T)\)) [24]. For this kind of circumstance in which data distribution changes little, we generally use transductive transfer learning methods.

The marginal and conditional distributions between domains are commonly adapted via transductive transfer learning techniques [25], [26]. It means reducing the distance between two distributions:

\[
D_{\text{dis}}(\mathcal{D}_S, \mathcal{D}_T) \approx D_{\text{dis}}(p(x_S), p(x_T)) + D_{\text{dis}}(p(y_S|x_S), p(y_T|x_T))
\]

where \(x_S\) and \(x_T\) are the sample features of the source and target domains; \(y_S\) and \(y_T\) are the labels of the source and target domains.

However, most of the existing transductive transfer learning methods are applied to classification problems, and there is little discussion on the transfer of DRL. The problem discussed in this paper is the characteristic transfer of voltage control. The difference between sample characteristics is mainly reflected in the differences between marginal distribution and conditional distribution. Balanced distribution adaptation (BDA) [27] is used to flexibly counterbalance discrepancies between the two distributions based on each particular task. It deploys a balancing factor \(\sigma\) to make use of the various distributions’ relative importance:

\[
D_{\text{dis}}(\mathcal{D}_S, \mathcal{D}_T) \approx (1 - \sigma)D_{\text{dis}}(p(x_S), p(x_T)) + \sigma D_{\text{dis}}(p(y_S|x_S), p(y_T|x_T))
\]  

(18)

where \(\sigma \in [0, 1]\).

Maximum mean discrepancy (MMD) [28] can be used to assess two distribution differences to calculate the distance in (14). The target domain \(\mathcal{D}_T\), on the other hand, lacks labels and hence cannot assess the conditional distribution \(p(y_T|x_T)\). To estimate, the class conditional distribution \(p(x_T|y_T)\) might be employed to approximate \(p(y_T|x_T)\). A base predictor trained on \(\mathcal{D}_S\) is utilized to compute the soft labels of \(\mathcal{D}_T\) to calculate \(p(x_T|y_T)\). Because these soft labels may not be precise enough, they are improved repeatedly. As a result, (18) is rewritten as

\[
D(\mathcal{D}_S, \mathcal{D}_T) \approx (1 - \sigma) \left\| \frac{1}{n} \sum_{i=1}^{n} x_{S_i} - \frac{1}{m} \sum_{j=1}^{m} x_{T_j} \right\|_H^2 + \sigma \sum_{c=1}^{C} \left\| \frac{1}{n_c} \sum_{x_{S_i} \in D^{(c)}_S} x_{S_i} - \frac{1}{m_c} \sum_{x_{T_j} \in D^{(c)}_T} x_{T_j} \right\|_H^2
\]

(19)

where \(c \in \{1, 2, \cdots, C\}\) is the class label; \(n\) and \(m\) are the number of samples in the source/target domain; \(D^{(c)}_S\) and \(D^{(c)}_T\) are the samples of label \(c\) in the source and target domain, respectively; \(n_c\) and \(m_c\) are the number of samples belonging

![Fig. 3. Different transfer learning methods for two circumstances.](image-url)
to $D_S^{(c)}$ and $D_T^{(c)}$; $\mathcal{H}$ is the reproducing kernel Hilbert space. For the prediction problem in this paper, the predicted value is divided into several sections, and each section is classified into one class. MADDPG output actions can be divided into three categories, in which [0, 1], [1, 2], and [2, 3] correspond to $VR$, $TC$, and $CR$, respectively.

Through matrix tricks and regularization, the optimization problem can be formed as follows:

$$\begin{align*}
\min_{A} & \quad \text{tr}(A^T X ((1 - \sigma)M_0 + \sigma \sum_{c=1}^C M_c) X^T A) + \omega \| A \|^2_F \\
\text{s.t.} & \quad A^T X H X^T A = I, 0 \leq \sigma \leq 1
\end{align*}$$

(20)

where $\omega$ is the regulation parameter; $\| \cdot \|^2_F$ is the Frobenius norm; $X$ is the input data matrix of $x_S$ and $x_T$; $A$, $I$ and $H$ are transformation matrix, identity matrix and centering matrix, respectively; $H = I - (1/n)11^T$; $M_0$ and $M_c$ are MMD matrices, which can be constructed as follows:

$$(M_0)_{ij} = \begin{cases} 
1/n^2, & x_i, x_j \in D_S \\
1/m^2, & x_i, x_j \in D_T \\
1/mn, & \text{otherwise}
\end{cases}$$

(21)

$$(M_c)_{ij} = \begin{cases} 
1/n^2, & x_i, x_j \in D_S^{(c)} \\
1/m^2, & x_i, x_j \in D_T^{(c)} \\
1/mnc, & \left\{ \begin{array}{l}
x_i \in D_S^{(c)}, \\
x_j \in D_T^{(c)}
\end{array} \right. \\
0, & \text{otherwise}
\end{cases}$$

(22)

The generalized Eigen decomposition is computed after establishing the Lagrange function for (20):

$$X \left( (1 - \sigma)M_0 + \sigma \sum_{c=1}^C M_c \right) X^T + \lambda I = X H X^T A \Phi$$

Equation (23) is solved to get the optimal transformation matrix $A$. The process of balanced distribution adaptation for AVC is summarized in Algorithm 1. The source label vector $\rho_S$ can be calculated based on Section II. According to the above steps, we can get

$$p(A^T X_S) = p(A^T X_T)$$

(24)

That is to say, the consistency of sample distribution between source and target domain is realized through the mapping of $A$.

C. Transfer Learning of Various Power Grids

For various power grids, although the source task and target task are the same, the structure changes significantly, resulting in feature space $X_S \neq X_T$ and label space $Y_S \neq Y_T$. If only using distribution adaption described in Section IV-B, the accuracy of feature transformation may not be high, which will lead to decrease of the effect. In this case, it is necessary to introduce parameter-based transfer learning, which retransfers the neural networks after transferring parameters. In addition to the parameter transfer of deep neural network, considering the constrained multi-agent DRL applied in this paper, the optimal policy of each agent can be transferred. Therefore, this paper transfers the trained network parameters and reinforcement learning policies.

First, transfer the trained network parameters. Pre-trained network weights are frequently used to initialize the weights of target networks rather than random initialization, which is called fine-tuning [29]. In this paper, we use the fine-tune method to freeze the partial layers of the pre-trained model (which are usually the layers close to the input), and train the remaining layers (which are usually the layers close to the output) with a small learning rate.

Secondly, transfer reinforcement learning policies. The specific steps of policy transfer [30] are shown in Algorithm 2. Moreover, the key of transfer learning is finding the $\rho$ which can map the source decision list $\xi_S$ to target decision list $\xi_T$. A homomorphism $h = \langle f, g \rangle |_{s \in S_T} [31]$ from target $M_T$ to source $M_S$ can be leveraged to generate a random policy $\pi$ in the target $M_T$ provided with a random policy $\pi_S$ in a source Markov game $M_S$, where $f : S_T \rightarrow S_S$.

Algorithm 1: Balanced Distribution Adaptation for AVC in Similar Power Grids

Input: Source and target feature matrix $X_S$ and $X_T$, source label vector $r_S$, dimension $d$, balancing factor $\sigma$, regulation parameter $\omega$

1. Train a base network on $X_S$ and apply prediction on $X_T$ to get soft labels $\hat{r}_T$. Build $X = [X_S, X_T]$.
2. Initialize $M_0$ and $M_c$ with (21) and (22).
3. repeat
   1. Solve the Eigen decomposition of (23) and utilize the smallest eigenvectors to construct $A$.
   2. Train a classifier $f$ on $\{A^T X_S r_S\}$
   3. Update the soft labels of $D_t$: $r_T = f_T(A^T X_T)$
   4. Update matrix $M_c$ by using (22).
4. until Convergence;

Output: Transformation matrix $A$

Algorithm 2: Policy Transfer for AVC in Various Power Grids

Input: Source and target domain and target

1. Learn a policy ($\pi : S \rightarrow A$) in the source domain.

   Following the learnt policy, the agent detects consecutive interactions with the environment in the form of ($S, A$) pairs when training is completed.

2. Create a decision list ($\xi_S : S \rightarrow A$) that outlines the source policy.

   Following data collection, a rule learner is utilized to abstract information in order to depict the learnt policy.

3. Update the decision list suitable for the target domain ($\rho(\xi_S) = \xi_T$).

   The learnt decision list is converted by Eq. (25) allowing it to be used in a target task with the states and actions different from the source task.

Output: Target policy $\xi_T$
projects states every block in $M_T$ to a specific state in $M_S$ and $g_s : \mathcal{A}_s \to \mathcal{A}_{f(s)}$ projects actions in the state $s$ to actions in the state $f(s)$. Define $g_s^{-1}(a_S) \subseteq \mathcal{A}_s$ as the collection of target actions that correspond to the same source action $a_S \subseteq \mathcal{A}_{f(s)}$. Then, for each $a \in g_s^{-1}(a_S)$ and $s \in \mathcal{S}_T$, the policy is

$$\pi_T(s, a) = \frac{\pi_S(f(s), a_S)}{|g_s^{-1}(a_S)|}$$  \tag{25}$$

$f$ and $g_s$ can be calculated according to the BDA introduced in Section IV-B. Based on (25), the mapping $\rho$ is formed. By Algorithm 2, an optimal learned policy in the source domain is transformed into an option policy in the target domain.

V. CASE STUDY

A. Experiment Setup

The proposed method’s verification systems include two IEEE systems (IEEE 14-bus system and IEEE 39-bus system) and two real-world power grids (Liaoning power grid and Heilongjiang power grid in China). The experiment was implemented on a high-performance server. TensorFlow (an open-source package) is used to write the code in Python. All power data are the standard unit values with a 100 MVA power base value, and voltage amplitude is the standard unit value.

Capacitors and reactors are incorporated into low-voltage buses in the IEEE 14-bus and IEEE 39-bus systems to carry more load and increase the range of voltage regulation. Generators and loads are randomly altered between 0.7 and 1.2 times, and capacitors/reactors’ switching conditions are adjusted concurrently to provide 5000 sets of power flow data, all based on the system’s initial power flow. The training set consists of 4000 data sets, while the testing set consists of 1000 data sets. And $V_{\text{low}}^\text{ref}$, $V_{\text{ref}}$ and $V_{\text{up}}^\text{ref}$ are set as 1.0 p.u., 1.05 p.u., and 1.1 p.u., respectively. For the transfer circumstance of systems with the same structure but distinct topological relations/parameters, five lines are removed from the original Liaoning power grid, and 10 generators are added. According to the administrative division of the real-world power grid, the DRL has four agents. To sum up, the transfer circumstances in Table II can be formed.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>IEEE system</th>
<th>Real-world power grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same system with distinct structures</td>
<td>14-bus $\rightarrow$ 14-bus</td>
<td>Liaoning $\rightarrow$ Liaoning</td>
</tr>
<tr>
<td>Various systems</td>
<td>14-bus $\rightarrow$ 39-bus</td>
<td>Liaoning $\rightarrow$ Heilongjiang</td>
</tr>
<tr>
<td>Voltage control range</td>
<td>[0.95, 1.05]</td>
<td>[1.0, 1.1]</td>
</tr>
</tbody>
</table>

B. Experimental Results

1) IEEE Systems

For the transfer circumstance of systems with the same structure but distinct topological relations/parameters, the distribution of sample parameters of IEEE 14-bus system after topological relations/parameters change has changed, as can be seen in Fig. 5. The voltage of node 3 in Fig. 5 increases significantly after topology relations/parameters change. Through the BDA built in Section IV-B, the MMD distance in Fig. 6 between the changed sample and the original sample is gradually reduced so the samples after structure change can adapt to the original model.

The constraint satisfying ratio (CSR) of the transferred samples assessed on the original model is 5.7% greater than before transfer, as seen in Table III, and control time and cost are significantly decreased. Control cost and control time denote the action cost and time to satisfy constraints, respectively; training time is the period from training to algorithm convergence. The electricity price is $40 / \text{MWh}$, and the per action cost of the device is $0.1.

The rewards with and without transfer learning during iteration for the IEEE 39-bus system are illustrated in Fig. 7,
the control process, the variation range of each node voltage is small, which indicates the constrained multi-agent DRL can effectively limit the voltage variation range. Moreover, node voltage can be adjusted to the constrained range in a few control steps, which is attributed to the integration of domain knowledge.

In Fig. 9, (a), (b), (c), and (d) show system-wide voltage magnitudes under heavy load and light load before and after control in the IEEE 39-bus system, respectively. It can be

and performance outcomes are reported in Table III, for the transfer circumstance of various power grids. Without transfer learning, the rewards of three agents continue to rise during the training iteration and gradually converge after 1500 rounds. Eventually, samples that meet the limitation approach 96.2%, demonstrating the suggested technique can optimize the voltage in the 39-bus system. Samples that fulfill the constraint are stable at around 98.1% after 1000 episodes with transfer learning, and the training time of the transferred samples on the initial model is 3.4 hours less than before transfer, reflecting transfer learning can not only speed up learning but also improve its effectiveness.

Figure 8 shows the voltage changes of five nodes in the IEEE 39-bus system during the control process. It can be

and performance outcomes are reported in Table III, for the transfer circumstance of various power grids. Without transfer learning, the rewards of three agents continue to rise during the training iteration and gradually converge after 1500 rounds. Eventually, samples that meet the limitation approach 96.2%, demonstrating the suggested technique can optimize the voltage in the 39-bus system. Samples that fulfill the constraint are stable at around 98.1% after 1000 episodes with transfer learning, and the training time of the transferred samples on the initial model is 3.4 hours less than before transfer, reflecting transfer learning can not only speed up learning but also improve its effectiveness.

Figure 8 shows the voltage changes of five nodes in the IEEE 39-bus system during the control process. It can be

and performance outcomes are reported in Table III, for the transfer circumstance of various power grids. Without transfer learning, the rewards of three agents continue to rise during the training iteration and gradually converge after 1500 rounds. Eventually, samples that meet the limitation approach 96.2%, demonstrating the suggested technique can optimize the voltage in the 39-bus system. Samples that fulfill the constraint are stable at around 98.1% after 1000 episodes with transfer learning, and the training time of the transferred samples on the initial model is 3.4 hours less than before transfer, reflecting transfer learning can not only speed up learning but also improve its effectiveness.

Figure 8 shows the voltage changes of five nodes in the IEEE 39-bus system during the control process. It can be

and performance outcomes are reported in Table III, for the transfer circumstance of various power grids. Without transfer learning, the rewards of three agents continue to rise during the training iteration and gradually converge after 1500 rounds. Eventually, samples that meet the limitation approach 96.2%, demonstrating the suggested technique can optimize the voltage in the 39-bus system. Samples that fulfill the constraint are stable at around 98.1% after 1000 episodes with transfer learning, and the training time of the transferred samples on the initial model is 3.4 hours less than before transfer, reflecting transfer learning can not only speed up learning but also improve its effectiveness.

Figure 8 shows the voltage changes of five nodes in the IEEE 39-bus system during the control process. It can be

and performance outcomes are reported in Table III, for the transfer circumstance of various power grids. Without transfer learning, the rewards of three agents continue to rise during the training iteration and gradually converge after 1500 rounds. Eventually, samples that meet the limitation approach 96.2%, demonstrating the suggested technique can optimize the voltage in the 39-bus system. Samples that fulfill the constraint are stable at around 98.1% after 1000 episodes with transfer learning, and the training time of the transferred samples on the initial model is 3.4 hours less than before transfer, reflecting transfer learning can not only speed up learning but also improve its effectiveness.
seen that under heavy loads, voltage magnitudes of the whole system are low, and plenty of voltages are lower than 0.95; under light load, voltage magnitudes of the whole system are high, and some of the node voltages are higher than 1.05. No matter under heavy or light load, voltages of the whole system can be controlled in the range of [0.95, 1.05]. Table IV shows test results under different load intervals. Load ratio denotes the ratio of the load for the current sample to original power flow. It can be seen when load ratio is 0.9∼1.0, the CSR is the highest. When the load ratio is close to 1, the number and degree of voltage off-limits are small, and difficulty of control is low. When the load is too low or too high, the CSR decreases and control cost and control time increase, but most samples can still be controlled within the constraint range.

### TABLE IV

<table>
<thead>
<tr>
<th>Load ratio</th>
<th>CSR</th>
<th>Control cost ($)</th>
<th>Control time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7~0.8</td>
<td>97.9%</td>
<td>5.9</td>
<td>2.8</td>
</tr>
<tr>
<td>0.8~0.9</td>
<td>98.9%</td>
<td>5.3</td>
<td>1.8</td>
</tr>
<tr>
<td>0.9~1.0</td>
<td>99.6%</td>
<td>4.6</td>
<td>1.5</td>
</tr>
<tr>
<td>1.0~1.1</td>
<td>98.2%</td>
<td>5.4</td>
<td>2.3</td>
</tr>
<tr>
<td>1.1~1.2</td>
<td>96.4%</td>
<td>6.1</td>
<td>3.0</td>
</tr>
</tbody>
</table>

The comparison of different schemes is shown in Table V. ’n full connected net+m fine-tune’ refers to that neural network has n full connected layers, and the first m-layer networks are frozen during fine-tuning.

### TABLE V

<table>
<thead>
<tr>
<th>Method</th>
<th>CSR</th>
<th>Control cost ($)</th>
<th>Control time (min)</th>
<th>Retraining time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3Full connected net+1 fine-tune</td>
<td>97.2%</td>
<td>5.6</td>
<td>2.2</td>
<td>6.8</td>
</tr>
<tr>
<td>3Full connected net+2 fine-tune</td>
<td>95.5%</td>
<td>6.3</td>
<td>3.1</td>
<td>4.8</td>
</tr>
<tr>
<td>4Full connected net+2 fine-tune</td>
<td>98.1%</td>
<td>5.5</td>
<td>2.1</td>
<td>5.2</td>
</tr>
<tr>
<td>Without reward of voltage deviation number</td>
<td>97.5%</td>
<td>6.1</td>
<td>2.8</td>
<td>6.9</td>
</tr>
<tr>
<td>Without reward of control cost</td>
<td>98.2%</td>
<td>6.8</td>
<td>3.5</td>
<td>5.7</td>
</tr>
<tr>
<td>Without voltage control strategy</td>
<td>94.4%</td>
<td>7.7</td>
<td>4.6</td>
<td>8.2</td>
</tr>
</tbody>
</table>

It can be seen that compared with other schemes, the performance of “4 full connected net+2 fine-tune” is best. For the scheme of “3 full connected net+2 fine-tune”, due to a few layers of neural networks for retraining, the neural network is overfitted, making the effect of a testing set poor. In addition, rewards without the number of voltage deviation nodes can lead to increased control time and training time, but CSR is good. If there is no reward for control cost, CSR will increase slightly, but control cost and control time will increase considerably. Therefore, adding the number of voltage deviation nodes and control cost to the reward is conducive to improving learning performance. When the voltage control strategy is removed, the whole system’s performance will be greatly reduced, especially training time increases sharply, which indicates the control strategy has an outstanding contribution to improving learning efficiency.

2) **Real-world Power Grids**

The MMD distance between the sample characteristics of the Liaoning power grid after the structure change and of the original sample is gradually decreased, similar to the IEEE systems, for the transfer circumstance of distinct structures of Liaoning power grid, as depicted in Fig. 10. Thus, the sample after the structure change can adapt to the original model. The effect of the transferred samples examined on the original model is 14.6% larger than before transfer, as seen in Table V.

The rewards with and without transfer learning during iterative process for the Heilongjiang power grid are presented in Fig. 11 for the transfer circumstance of various power grids, and performance outcomes are given in Table VI. Without transfer learning, the rewards of four agents continue to rise after 5000 repetitions, comparable to the 39-bus system. Finally, 92.9% of the samples fulfill the restriction. After 2000...
iterations, the samples that fulfill the restriction can approach 94.6% via transfer learning. The number of samples fulfilling constraint falls as compared to the 39-bus system. Nonetheless, it represents major time-saving in terms of training. Therefore, applying transfer learning to different real-world power grids is very significant in shortening training time.

In Fig. 12, (a), (b), (c), and (d) show system-wide voltage magnitudes under heavy load and light load before and after control in the Heilongjiang power grid, respectively. Table VII shows test results under different load intervals. It can be seen that under heavy load, voltage magnitudes of the whole system are low, and about a third of the node voltages are lower than 1.0; under light load, voltage magnitudes of the whole system are high, and some of the node voltages are higher than 1.1, few node voltages even reach 1.2. Similar to the IEEE 39-bus system, as shown in Table VII, the CSR is highest when the load ratio is 0.9–1.0. No matter under heavy load or light load, voltage magnitudes of the whole system can be controlled to be in the range of [1.0, 1.1] for most samples.

![System-wide voltage magnitudes under heavy load and light load before and after control in the Heilongjiang power grid.](image)

**Fig. 12.** System-wide voltage magnitudes under heavy load and light load before and after control in the Heilongjiang power grid.

<table>
<thead>
<tr>
<th>Load ratio</th>
<th>CSR (%)</th>
<th>Control cost ($)</th>
<th>Control time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7–0.8</td>
<td>93.2%</td>
<td>16.3</td>
<td>9.5</td>
</tr>
<tr>
<td>0.8–0.9</td>
<td>94.8%</td>
<td>15.2</td>
<td>9.3</td>
</tr>
<tr>
<td>0.9–1.0</td>
<td>95.8%</td>
<td>14.6</td>
<td>8.8</td>
</tr>
<tr>
<td>1.0–1.1</td>
<td>94.2%</td>
<td>15.0</td>
<td>9.1</td>
</tr>
<tr>
<td>1.1–1.2</td>
<td>93.6%</td>
<td>16.1</td>
<td>9.6</td>
</tr>
</tbody>
</table>

**TABLE VII**

**TEST RESULTS OF DIFFERENT LOAD INTERVALS FOR HEILONGJIANG POWER GRID**

**C. Comparison and Analysis**

The proposed approach in this study is compared to five algorithms to demonstrate the advantages of the algorithm utilized in this work. These five algorithms include multi-agent proximal policy optimization (MAPPO), Q learning with mixed networks (QMIX), DDPG, PPO and soft actor-critic (SAC).

The MAPPO approach has a little worse impact than the MADDPG method, as seen in Fig. 13, because MADDPG has higher sampling efficiency when computing resources are limited. For QMIX, the CSR, cost control, and time control are not as good as MADDPG and MAPPO. Because QMIX is extended from Q learning, Q learning is completely based on offline policy learning, and the effect is not as good as DDPG and PPO. In terms of DDPG, PPO, and SAC of single-agent, CSR is much lower than of multi-agent algorithms. For the voltage control problem solved in this paper, the control variables are distributed and interactive, so it is difficult for a single agent to control all variables effectively. Even though SAC performs slightly better, it still lags behind the effect of multi-agent DRL.

![Comparisons of test results for this method and other algorithms.](image)

**Fig. 13.** The comparisons of test results for this method and other algorithms.

**VI. CONCLUSION**

To solve the problem of model-free voltage control methods, this paper proposes an AVC method for differential power grids based on transfer learning and deep reinforcement learning. A constrained multi-agent DRL-based AVC method is developed based on MADDPG and voltage control strategy. Two transfer learning paradigms are developed for use in two AVC transfer scenarios. In addition, the method’s efficacy is tested using two IEEE systems and real-world power grids.

The proposed approach shown in the case study can efficiently control voltage to satisfy the restriction range. Constrained multi-agent DRL not only can balance the mutual impact of control objects but also can limit action in the control process. Using domain knowledge, learning efficiency is enhanced. In addition, adding the number of voltage deviation nodes and control cost to the reward is conducive to improving learning performance. Distribution adaptation can project the data of the target domain to the same distribution as the source domain, and target samples can achieve good performance on the source model without any further training in the AVC transfer circumstance of systems with the same structure but different topological relations/parameters. Parameter-based transfer learning can enhance training speed and effect of the target task in the AVC transfer circumstance of various power grids.

**REFERENCES**

[1] C. Ren, Y. Xu, J. Zhao, R. Zhang and T. Wan, “A super-resolution perception-based incremental learning approach for power system volt-


