Transient Stability Preventive Control Based on Graph Convolution Neural Network and Transfer Deep Reinforcement Learning

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

***Abstract*—This study proposes a new transient-stability preventive control (TSPC) method based on graph convolutional neural networks (GCNN) and transfer deep reinforcement learning (DRL) to address non-convergence problems of traditional optimization algorithms and slow training speed of artificial intelligence algorithms for TSPC. First, a transient stability assessor (TSA) with GCNN is developed to assess current-power-flow state. Sensitivities of the transient-stability index relative to the generators are approximately calculated using TSA; generators with significant influence able to narrow action space are identified. Subsequently, the Markov decision-making process of TSPC is derived by introducing the process of TSPC. A DRL for TSPC is constructed by adding entropy to twin delayed deep deterministic policy gradient (TD3). Knowledge learned by TSA is transferred to DRL based on transfer learning, which improves learning efficiency. Finally, case studies based on the IEEE 39-bus system and an actual power grid prove the effectiveness of the proposed method. Comparisons performed with reference algorithms in literature demonstrate the proposed method has better performance in both control effect and speed.**

***Index Terms*—Deep reinforcement learning, graph convolution, preventive control, transfer, transient stability.**

NOMENCLATURE

A. Variables

$\boldsymbol{s}, \boldsymbol{a}$	State and action.
r	Reward.
π	Policy.
ε_{L}	Load level.
P_i	Active power of the ith generator.
n_{G}	Number of generators.
μ	Fault location.
$\boldsymbol{I}$	Inputs of neural network.
$\boldsymbol{o}$	Output of neural network.
$\boldsymbol{Y}$	Node admittance matrix.
$\boldsymbol{U}$	Eigenvector after decomposition.
λ	Eigenvalue.
$\boldsymbol{D}$	Vertex's degree matrix.
$\boldsymbol{H}$	Graph's adjacency matrix.
$S_{\mathrm{G}i}$	Sensibility of the ith actionable generator.
$\boldsymbol{\chi}^+, \boldsymbol{\chi}^-$	Positive and negative adjustment generator sequences.
α_k, β_k	kth generator of $\boldsymbol{\chi}^+$ and $\boldsymbol{\chi}^-$.
n_{A}	Number of actionable generator pairs.
$\Delta P_{\mathrm{Gpair}}$	Adjustment power of a generator action pair.
N^i_{uns}	Number of unstable generators after the ith fault.
n_{G}	Number of generators.
$TSI^j_{\mathrm{us}i}$	Amount of TSI of the ith unstable generator after the jth fault.
$\Delta P, \Delta Q$	Active and reactive power action quantities.
V	State value function.
Q	Action value function.
γ_t	Discount factor at time t.

B. Parameters

$\delta_{\max}$	Maximum rotor angle difference between any two generators.
θ	First ChebNet parameter.
$\boldsymbol{x}_{nm}$	Input parameters of each layer.
b_{nm}	Weights and biases of input parameters.
$C_{\mathrm{p}}, C_{\mathrm{Q}}$	Active and reactive power unit action costs.
λ_{R}	Reward coefficient.
γ_t	Discount factor at time t.
T	Time horizon.
α	Update coefficient.
ω_1, ω_2	Parameters of two critic networks.
θ_1, θ_2	Parameters of two actor networks.
τ	Proportion of update.
ε	Clipped random noises.
σ, c	Parameters of noise.
κ	Entropy coefficient.

C. Abbreviations

TSPC	Transient-stability preventive control.
GCNN	Graph convolution neural network.
TSA	Transient stability assessor.
TD3	Twin delayed deep deterministic policy gradient.
TSCOPF	Transient stability constrained optimal power flow.
SCR	Stability-constrained rescheduling.
IPM	Interior point method.
TSI	Transient stability index.

Manuscript received July 25, 2022; revised September 23, 2022; accepted November 1, 2022. Date of online publication January 25, 2023; date of current version February 13, 2023.

T. J. Wang (corresponding author, email: 18810303378@163.com) is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

Y. Tang is with the Laboratory of Power Grid Safety and Energy Conservation, China Electric Power Research Institute, Beijing 100192, China.

DOI: 10.17775/CSEEJPES.2022.05030

MDP	Markov decision-making process.
FCL	Fully connected layers.
DRL	Deep reinforcement learning.
AAS	Average adjustment steps.
CSR	Constraint satisfaction ratio.
ANN	Artificial neural network.
CNN	Convolutional neural network.

I. Introduction

IT is crucial to ensure stable and secure operation of power grids. However, modern power systems have become more complex and uncertain owing to vast usage of power electronics and increased penetration of renewable energy resources, which increases probability of system failure [1], [2]. Transient instability is one of the leading causes of large-scale accidents in power systems, and effective TSPC is the key to solving this problem [3]. TSPC controls operation states of the system to an operation point that satisfies the requirements of transient stability after the system is disturbed. The actual project realizes TSPC through manual adjustment. The process is to perform transient stability simulation considering $N-1$ contingencies for initial power flow ($N-1$ calculation) first, observe the instability state of power flow, and then control the generator output based on instability state. However, weak orientation and uncertain amounts in the control process may lead to turbid ebb and flow, and low control efficiency.

In previous studies, the TSPC problem was solved using two approaches: transient stability constrained optimal power flow (TSCOPF) and stability-constrained rescheduling (SCR). These studies developed TSCOPF from two directions. On the one hand, it improved numerical differentiation of TSCOPF with the interior point method (IPM), such as reduced space IPM [4], two-level parallel decomposition [5], and primal-dual Newton IPM [6]. On the other hand, transient-stability constraints were transformed into other forms, such as transient-sensitivity analysis [7], independent TSA [8], individual machine-equal area criterion [9], and energy-sensitivity method [10]. The commonly used solution methods include traditional accurate algorithms and intelligent algorithms. Reference [11] proposed a heuristic benders-decomposition-based algorithm to solve the TSCOPF problem. An improved chaotic electromagnetic field optimization algorithm was utilized in [12] to determine an optimal output of generating units, ensuring both economy and stability. Nevertheless, whichever TSCOPF model is used, it may suffer from non-convergence and local optimum problems for actual large-scale power grids due to complexity of its constraints. In terms of SCR, the authors in [13] provided a method based on trajectory sensitivity and extended equal-area criterion to reschedule power generation. Subsequently, a coordinated method for preventive generation rescheduling and corrective load shedding for TSPC was presented in [14] with trajectory sensitivity and extended equal area criterion, which can maintain power system transient stability under uncertain wind-power variation. However, control cost limits this method. Moreover, for both TSCOPF and CSR, the control strategy is constructed based on historical data, which can fail to respond to unknown changes in the real-time environment and lead to control failures.

Driven by recent advances in artificial intelligence, some researchers have conducted initial explorations using deep neural networks for TSPC. Reference [15] transferred energy functions into a series of differential equations and customized Hopfield neural network for constrained optimization. [16] proposed the TSCOPF model incorporating a Bayesian neural network. Moreover, the contingency-oriented XGBoost model was trained in [17] to represent the transient-security constraint in the TSCOPF, which has relatively fast calculation speed. For the three methods above, adding the learned neural network to TSCOPF reduces the computational difficulty to some extent; however, the computational complexity of the OPF remains extremely high. Especially for large-scale power grids, the solving processes may still not converge. A support-vector machine-based method for TSPC was proposed in [18] to solve this problem. By this means, sensitivities of the transient stability index (TSI) with respect to control variables are calculated and ranked to select control generators. Subsequently, the power-shifting amount of all control generators is calculated. However, control is costly and does not fully mine the information in the neural network.

Currently, some researchers have primarily applied DRL to predictive control. A DRL-based method was adopted in [19] to control security for different power-grid operation modes with uncertainties for the rotor angle stability problem of large-scale systems after UHVDC bipolar block. Additionally, a parallel deep reinforcement-learning model, that maps actions to a pair of generators and realizes parallel control of multi objectives, was established in [20] for security control. Under a multiagent environment, [21] utilized distributed deep deterministic policy gradient to learn its control policy through massive interactions with a grid simulator, which can control the system to a safe operating point. Although the DRL has certain advantages in terms of preventive control issues, the high complexity of TSPC environment may result in an extremely long training time. Transfer learning can reduce training time of transferring objects to some extent. Researchers in [22] established a mechanism of knowledge transfer across power flow snapshots for representative risky fault chain identification, and proposed strategies of transition and extension for accelerating computation based on previously learned knowledge. In [23], a transfer strategy was constructed to transfer partial network parameters of a well-trained neural network to construct the prediction model of a target wind farm. One trained dynamic security assessment in [24] was transferred to an unknown different but related fault by iteratively minimizing marginal and conditional distribution differences between trained data and unknown data.

Owing to extensive action space and high training cost of DRL for TSPC, this study develops a TSA using GCNN, which can reduce action space based on sensitivities calculated by the trained TSA and transfer the learned parameters to DRL. This study uses DRL combined with GCNN and transfer learning to realize TSPC, as shown in Fig. 1. In the figure, the upper part is the TSPC process. In the left dashed box of the lower part, trajectory sensitivities are calculated by the

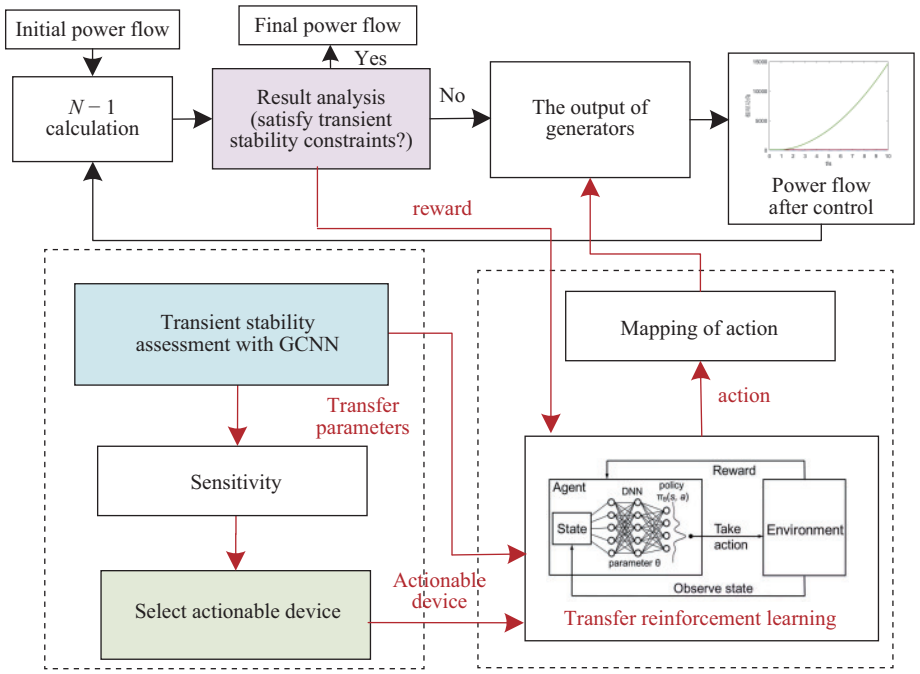

Fig. 1. Schematic diagram of TSPC based on GCNN and transfer DRL.

parameters of the learned GCNN, and actionable devices are located; in the right dashed box, neural network parameters are transferred to DRL, and power flow is controlled by DRL. This study considers only rotor angle instability. Contributions of this study are as follows.

1) Based on GCNN, a TSA for the current-power-flow state is established by fitting the relationship between TSI and output power of generators with neural networks. The TSA considers current state of the system and adds a graph structure of the system, which can evaluate transient-instability degree of the system after a fault, when the current-power-flow state and fault are known.

2) Sensitivity of each generator to TSI is approximately calculated using the learned TSA, and generators with significant influence are identified. Action space of DRL is narrowed, and control efficiency is improved by actioning the generators. This method uses the information learned from the neural network to constitute the relationship between the parameters of the neural network and the predicted outcome, which makes the neural network interpretable.

3) Based on the process of TSPC, state space, action space, policy, and reward function are constructed to form the Markov decision-making process (MDP) of TSPC. Based on TD3 with entropy, a DRL for TSPC is constructed.

4) Knowledge learned by TSA is transferred to DRL based on transfer learning. Partial neural networks are retrained, and power flow is controlled. This method makes full use of the learned information of neural networks and reduces training cost. Compared with traditional analytical/optimization-based approaches, the control speed of this method is advantageous. Besides, this method provides better comprehensive control effect relative to normal learning-based methods.

The remainder of this paper is organized as follows: TSA and actionable device selection are introduced in Section II. Section III presents a TSPC method based on transfer DRL. In Section IV, case studies demonstrate the effectiveness of the proposed method. Section V discusses related work and provides concluding remarks.

II. TSA and Actionable Device Selection Based on GCNN

A TSA with GCNN is first constructed to learn the relationship between power-flow state and transient stability. Subsequently, sensitivities of the TSI relative to generators are calculated, and actionable devices are found.

A. TSA Based on GCNN

1) Input and Output Parameters

Inputs of TSA, which include output power of each generator, current-fault situation, and load level, are the influencing factors of transient stability. Therefore, input parameters are the active power of each generator, fault location, and load level, as shown in the following formula.

$$\boldsymbol{I}(t) = [\varepsilon_{\mathrm{L}}, P_1, P_2, \cdots, P_{n_{\mathrm{G}}}, \mu] \tag{1}$$

where ε_{L} denotes load level; P_i is active power of the ith generator; n_{G} is the number of generators; μ is fault location.

TSI [25] is selected as the transient-stability evaluation criterion. TSI reflects maximum rotor angle difference in the transient process. Mathematical expression of TSI is as follows:

$$TSI = \frac{360 - \delta_{\max}}{360 + \delta_{\max}} \times 100\% \tag{2}$$

where $\delta_{\max}$ is the maximum rotor angle difference between any two generators.

After a fault, usually, not only one generator is unstable. To reflect the instability degree of the current system, the maximum TSI is calculated as shown in the following formula.

$$TSI_{\max} = \max\{TSI_1, TSI_2, \cdots TSI_{n_{\text{us}}}\} \tag{3}$$

where $TSI_{\max}$ is the maximum TSI of unstable generators; n_{us} is the number of unstable generators.

By calculating the TSI of current state, the output $\boldsymbol{o}(t)$ of the network is obtained by

$$\boldsymbol{o}(t) = TSI_{\max} \tag{4}$$

2) GCNN

A power system is a network composed of nodes and branches, which can be observed as a graph structure. This study introduces GCNN [26] to learn the graph structure of the system to thoroughly learn system information. Fig. 2 shows the schematic diagram of the GCNN application to the power system.

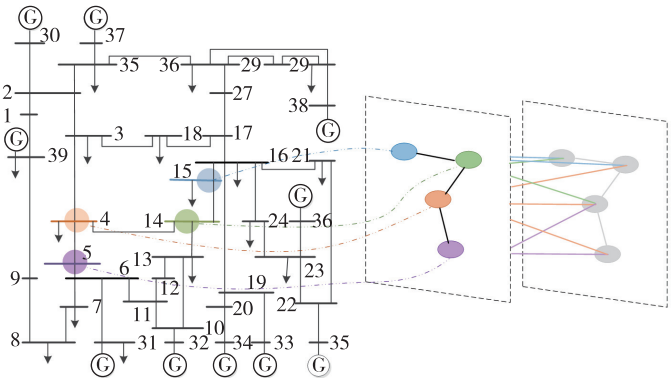

Fig. 2. The schematic diagram of the graph convolution.

Convolutional networks employing graph-structure data are called graph convolutional networks. Two inputs are used in a layer of the graph convolutional network. First is the feature input $\boldsymbol{I}(t)$ of each node. The other is the graph structure matrix, which is a normalized graph Laplacian matrix corresponding to the node admittance matrix $\boldsymbol{Y}$ of the power system.

On the graph, the convolution of $\boldsymbol{I}(t)$ with convolutional kernel g is expressed as follows.

$$(\boldsymbol{I}(t) * \boldsymbol{g})_{\text{G}} = \boldsymbol{U}((\boldsymbol{U}^{\text{T}}\boldsymbol{g}) \odot (\boldsymbol{U}^{\text{T}}\boldsymbol{I}(t))) \tag{5}$$

where $\boldsymbol{U}$ refers to the eigenvector obtained after decomposition, and the eigen-decomposition procedure is as follows:

$$\boldsymbol{Y}' = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1} = \boldsymbol{U}\begin{bmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_n\end{bmatrix}\boldsymbol{U}^{-1} \tag{6}$$

where $\boldsymbol{Y}'$ denotes the node admittance matrix after symmetric normalization; and $\lambda_1 \cdots \lambda_n$ are the eigenvalues of $\boldsymbol{Y}'$.

Reference [27] introduced a first-order approximation of ChebNet [28], and the graph convolution formula can be rewritten as

$$(\boldsymbol{I}(t) * \boldsymbol{g})_{\text{G}} = \theta(\boldsymbol{I}_n + \boldsymbol{D}^{-1/2}\boldsymbol{H}\boldsymbol{D}^{-1/2})\boldsymbol{I}(t) \tag{7}$$

where $\boldsymbol{D} = \text{diag}(d)$ represents the degree matrix (diagonal matrix) of the vertex; $\boldsymbol{H}$ denotes the adjacency matrix of the graph; and θ is the first ChebNet parameter.

$\boldsymbol{I}_n + \boldsymbol{D}^{-1/2}\boldsymbol{H}\boldsymbol{D}^{-1/2}$ is an eigenvalue matrix with a range of $(0, 2)$, which may lead to numerical instability and gradient explosion or gradient disappearance. Enabling a renormalization trick, let $\tilde{\boldsymbol{H}} = \boldsymbol{H} + \boldsymbol{I}_n$ and $\tilde{\boldsymbol{D}}_{ii} = \sum_j \tilde{\boldsymbol{H}}_{ij}$, and

$$\boldsymbol{I}_n + \boldsymbol{D}^{-1/2}\boldsymbol{H}\boldsymbol{D}^{-1/2} = \tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{H}}\tilde{\boldsymbol{D}}^{-1/2}. \tag{8}$$

3) TSA

A TSA is built based on the above input and output parameter settings and GCNN construction. Fig. 3 shows that input data is extracted from graph features through a graph-convolution layer and realize model learning through a fully connected layer. The graph-convolution layer is a single-layer neural network, and the fully connected layer can be selected as an appropriate number of layers according to the training effect. The TSA constructed in this paper is designed to calculate the sensitivity of the TSI relative to the generator output on the one hand, and to transfer the learned parameters to the DRL on the other hand, thus accelerating the training process of DRL.

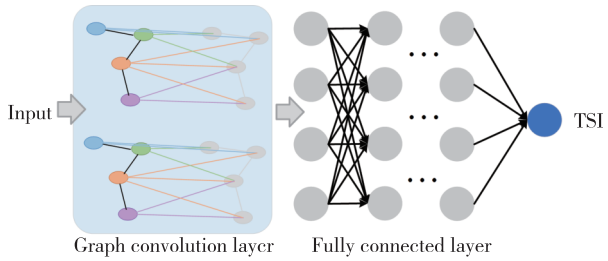

Fig. 3. TSA based on GCNN.

B. Actionable Device Selection Based on Sensitivity

Output of a fully connected layer can be expressed as follows.

$$\boldsymbol{o} = \sum_{m=1}^{M_n} \sigma \left(\omega_{nm} \sum_{m=1}^{M_{n-1}} \sigma\left(\omega_{(n-1)m}\boldsymbol{x}_{(n-1)m} + b_{(n-1)m}\right) + b_{nm}\right) \tag{9}$$

where $\boldsymbol{x}_{nm}$ represents input parameters of each layer; ω_{nm} and b_{nm} are the weights and biases of input parameters with respect to the mth neuron of the nth layer; σ denotes the activation function; N is the number of layers of neural network; Mn is the neuron number of each layer. Here, f_n is defined.

$$f_n(\boldsymbol{I}) = \sum_{m=1}^{M_n} \sigma \left(\omega_{nm} \sum_{m=1}^{M_{n-1}} \sigma(\omega_{(n-1)m}\boldsymbol{x}_{(n-1)m} + b_{(n-1)m}) + b_{nm}\right) \tag{10}$$

Therefore, (8) can be derived as

$$\boldsymbol{o} = f_n(\boldsymbol{I}) \tag{11}$$

and the derivation of f_n is:

$$\tilde{f}_n(\boldsymbol{I}) = \sum_{m=1}^{M_n} \sigma' \left(\omega_{nm} \sum_{m=1}^{M_{n-1}} \sigma(\omega_{(n-1)m} \boldsymbol{x}_{(n-1)m} + b_{(n-1)m}) + b_{nm} \right) \tag{12}$$

The partial derivative of output to input is as follows.

$$\frac{\mathrm{d}\boldsymbol{o}}{\mathrm{d}\boldsymbol{I}_i} = \prod_{n=1}^{N} \omega_{ni} \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{I}) \tag{13}$$

By adding a convolutional layer, output is expressed as follows.

$$y = f_n(\tilde{\boldsymbol{D}}^{-1/2} \tilde{\boldsymbol{H}} \tilde{\boldsymbol{D}}^{-1/2} \boldsymbol{I}) \tag{14}$$

Therefore, the partial derivative of output to input is

$$\frac{\mathrm{d}\boldsymbol{o}}{\mathrm{d}\boldsymbol{I}_i} = \tilde{\boldsymbol{D}}^{-1/2} \tilde{\boldsymbol{H}} \tilde{\boldsymbol{D}}^{-1/2} \prod_{n=1}^{N} \omega_{ni} \prod_{n=2}^{N} \tilde{f}_n(\boldsymbol{I}) \tag{15}$$

Value of the partial derivative can be used as the sensitivity of an actionable generator.

$$S_i = \frac{\mathrm{d}\boldsymbol{o}}{\mathrm{d}\boldsymbol{I}_i} \tag{16}$$

where S_i is the sensitivity of the ith generator with respect to TSI.

In this paper, generators with relatively high sensitivity are used as actionable devices, while generators with relatively low sensitivity are generally not used. To find a generator with relatively high sensitivity, generators are classified and sorted according to the amount of S. For current transient stability samples, positive and negative adjustment generator sequences are as follows.

$$\begin{cases} \boldsymbol{\chi}^+ = \{\alpha_k | S_{\alpha_k} > S_{\alpha_{k+1}}, S_{\alpha_k} > 0, S_{\alpha_{k+1}} > 0\} \\ \boldsymbol{\chi}^- = \{\beta_k | |S_{\alpha_k}| > |S_{\alpha_{k+1}}|, S_{\alpha_k} < 0, S_{\alpha_{k+1}} < 0\} \end{cases} \tag{17}$$

where $\boldsymbol{\chi}^+$ and $\boldsymbol{\chi}^-$ refer to positive and negative adjustment generator sequences according to the amount of S; α_k and β_k are the kth generator of $\boldsymbol{\chi}^+$ and $\boldsymbol{\chi}^-$, respectively. Thus, actionable generator pairs $\boldsymbol{A}_{\mathrm{G}}$ are formed, as shown in formula (18).

$$\boldsymbol{A}_{\mathrm{G}} = \{\{\alpha_1, \beta_1\}, \cdots, \{\alpha_{n_{\mathrm{A}}}, \beta_{n_{\mathrm{A}}}\}\} \tag{18}$$

where n_{A} is the number of actionable generator pairs. Each generator pair is chosen in turn according to the priority in (18) and the action amount is calculated in (20). If power after action does not exceed the generator capacity limit, the action is performed; otherwise, the next generator pair is chosen.

III. TSPC Based on Transfer DRL

A. The Construction of MDP

The MDP is a sequential decision-making mathematical model that is used to simulate random policies achieved by intelligent agents [29]. The TSPC process, as described in Section I, is a rescheduling decision-making process based on real-time feedback. The MDP can be used to depict it. State, action, policy, and reward are the primary factors in MDP. The corresponding factors of TSPC are described as follows.

- State (state space)

State space represents the observable quantity of the current object, which is consistent with the input parameters of TSA.

$$\boldsymbol{s} = \boldsymbol{I}(t) = [\varepsilon_{\mathrm{L}}, P_1, P_2, \cdots, P_{n_{\mathrm{G}}}, \mu] \tag{19}$$

- Action (action space)

For actionable devices, DRL determines the amount of action. Continuous action is mapped to the output-power change of actionable generator pairs based on the following mapping relationship. Action of DRL is mapped to the change amount of TSI, and the change amount of the actionable generator pair is calculated through sensitivity according to (16). Action space and mapping process are as follows.

$$a = [0, 1] \rightarrow \Delta TSI = TSI_{\max} * a \xrightarrow{S_{\mathrm{G}i}} \Delta P_{\mathrm{Gpair}} \tag{20}$$

where $\Delta P_{\mathrm{Gpair}}$ is the adjustment power of an actionable generator pair. Combining (20) and (18), the action amount for each generator pair is calculated, taking into account the capacity limits of the generators. If the generator pair with higher priority has reached its capacity limit, the action continues with the generator pair of the next priority. The mapping process is implemented in one step. First, action $\boldsymbol{a}$ is determined by the DRL. Then through the mapping in (20), the adjustment amount of TSI is the product of $\boldsymbol{a}$ and the maximum TSI, which can be obtained by GCNN. Finally, the change amount of the actionable generator pair $\Delta P_{\mathrm{Gpair}}$ is calculated according to the sensibility. For each power system state, the actionable generator pair is selected by (18). After each action of DRL, the power flow state changes, and the corresponding actionable generator pair needs to be re-selected.

- Policy

The policy π of MDP is given based on the state, which is the conditional probability distribution p of action, expressed as follows.

$$\pi(a|s) = p(a|s) \tag{21}$$

- Reward

A particular reward should be supplied according to the stability requirement to lead the action to make power flow fulfill the transient-stability constraint. The following is the specific reward design.

After an action of DRL, an $N-1$ calculation needs to be executed so that performance of the DRL action can be evaluated. N sets of transient-stability samples can be produced after the $N-1$ calculation. Moreover, the number of unstable generators in each sample can be determined based on

the instability condition of these N sets of samples. Therefore, the total number of unstable generators is

$$N_{\mathrm{Zuns}} = \sum_{i=1}^{N} N_{\mathrm{uns}}^{i} \tag{22}$$

where N_{uns}^{i} refers to the number of unstable generators after the ith fault. Total number of unstable generators after each action is incorporated into the reward function to demonstrate the instability state of power flow, which is expressed as follows.

$$r_1 = -N_{\mathrm{Zuns}}/Nn_{\mathrm{G}} \tag{23}$$

where n_{G} is the number of generators.

The quantity of TSI should also be considered to represent the current level of instability. The reward is set accordingly as follows.

$$r_2 = \frac{1}{N}\sum_{i=1}^{N} \min_{j=1,2,\cdots,N_{\mathrm{uns}}^{i}} TSI_{\mathrm{us}i}^{j} \tag{24}$$

where $TSI_{\mathrm{us}i}^{j}$ is the amount of TSI of the jth unstable generator after the ith fault.

Furthermore, by considering control cost, the reward can be determined as follows.

$$r_3 = -\lambda_{\mathrm{R}} C_{\mathrm{P}} \Delta P \tag{25}$$

where C_{p} is the unit action costs of active power; ΔP is the action quantities of active power; λ_{R} is the reward coefficient. In this study, λ_{R} is fixed as 0.1 to maintain it in the same range as previous rewards.

In summary, the total reward r is

$$r = r_1 + r_2 + r_3 \tag{26}$$

This study takes current power flow state as the state of MDP and maps the action of MDP into the power adjustment amount of the generators, as described in (19) and (20). The relevant resultant parameters after $N-1$ calculation (as described in (22)–(25)) are then used as the rewards of MDP, thus converting the TSPC problem to an MDP.

The DRL's purpose is to discover a policy that maximizes the expected discounted reward

$$\max_{\pi} \mathop{\mathbb{E}}_{\substack{a_t\sim\pi_i \\ s_{t+1}\sim p(s_{t+1}|s_t,a_t)}} \left[\sum_{t=0}^{T} \gamma_t r_t\right] \tag{27}$$

where γ_t is a discount factor at time t; and T is the time horizon.

The state-value function $V(s)$ and action-value function $Q(s,a)$ are two important functions that can be defined as

$$V(s) = \mathop{\mathbb{E}}_{\substack{a_t\sim\pi_i \\ s_{t+1}\sim p(s_{t+1}|s_t,a_t)}} \left[\sum_{t=0}^{T} \gamma_t r_t | s_0 = s\right] \tag{28}$$

$$Q(s,a) = \mathop{\mathbb{E}}_{\substack{a_t\sim\pi_i \\ s_{t+1}\sim p(s_{t+1}|s_t,a_t)}} \left[\sum_{t=0}^{T} \gamma_t r_t | s_0 = s, a_0 = a\right] \tag{29}$$

B. DRL for TSPC

A Bellman equation of state-value function Q is constructed based on the above formulation of MDP.

$$Q_k(s,a) = Q_{k-1}(s,a) + \frac{1}{N}[r(s') + \max_{a'} Q_{k-1}(s',a') - Q_{k-1}(s,a)] \tag{30}$$

where s, s' and a, a' are states and actions of the current and subsequent moments, respectively.

After finding an actionable device through the TSA, action of DRL is mapped to the output-power change of the generator, and action is continuous. This study establishes a DRL model based on TD3 [30]. Policy is updated in TD3 using the deterministic policy gradient. (31) shows that given current-policy parameters ϕ, then ϕ_{approx} refers to the parameters of the actor update induced by the maximization of the approximate critic $Q_\theta(s,a)$. In deterministic policy gradients, the ϕ_{approx} are overestimated under some basic assumptions. TD3 uses the clipped double Q-learning and delayed update to avoid overestimation.

$$\phi_{\mathrm{approx}} = \phi + \alpha \mathbb{E}_{s\sim p_\pi}\left[\nabla_\phi \pi_\phi(s) \nabla_a Q_\theta(s,a)|_{a=\pi_\phi(s)}\right] \tag{31}$$

where α is the update coefficient.

At the same time, considering the uniformity demand of action space in the process of TSPC, entropy is added to the loss function to explore the space evenly [31]. Thus, we make the following improvements on DRL.

1) Clipped double Q-learning: In double Q-learning, two networks independently select actions and estimate Q-values. Clipped double Q-learning favors underestimation bias over overestimation bias, which is difficult to spread during training:

$$\begin{cases} y_1 = r + \gamma \min\limits_{i=1,2} Q_{\omega 2}(s', \pi_{\theta_1}(s')) \\ y_2 = r + \gamma \min\limits_{i=1,2} Q_{\omega 1}(s', \pi_{\theta_2}(s')) \end{cases} \tag{32}$$

where $\gamma \in [0,1]$ is the discount factor; ω_1 and ω_2 are the parameters of two critic networks; θ_1 and θ_2 are the parameters of two actor networks.

2) Delayed updating of target and policy networks: TD3 updates policy at a lower frequency to decrease variation compared to Q-function. After repeated updates, the policy network remains the same until the value error is minimal enough [32]. Consequently, the delayed updates are as follows.

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \tag{33}$$

where θ and θ' are the parameters of policy network before and after update; τ is the proportion of update.

3) Target policy smoothing: on the value function, TD3 introduces a smoothing regularization approach that adds a small amount of clipped random noise to the selected action and averages over mini-batches [30], [33], which can reduce bias in the estimation of the value function. Therefore, the value function is

$$\begin{cases} V(s') = r + \gamma Q_\omega(s', \pi_\theta(s') + \varepsilon) \\ \varepsilon \sim \mathrm{clip}(N(0,\sigma), -c, +c) \end{cases} \tag{34}$$

where ε denotes clipped random noise; σ and c are the parameters of noise.

4) Adding noise to policy and entropy to reward: to increase the randomness of policy the policy is added with noise as the following formula.

$$a \sim \tilde{\pi} = \pi(s) + \varepsilon, \quad \varepsilon \sim N(0, \sigma) \tag{35}$$

where ε is a noise; $\tilde{\pi}$ is the policy with noise; $N(0, \sigma)$ refers to a normal distribution with expectation of 0 and variance of σ.

To further measure and improve the randomness of the action, entropy $H(\tilde{\pi}) = -\log(\pi(s)+\varepsilon)$ can be incorporated into the reward, which enables policy to create positive feedback on randomness. Consequently, the $V(s)$ is rewritten as

$$V(s) = Q(s, a) - \kappa \log(\pi(s) + \varepsilon) \tag{36}$$

where κ is the coefficient of entropy.

C. Transfer DRL

Transfer learning [34] is introduced to transfer the information learned by TSA to DRL to fully exploit the learned information of neural networks. Transfer learning makes the training of the target task more flexible and efficient by applying experience learned from the source task to target task. Because the parameters of TSPC are closely related to TSA, it belongs to transfer learning of related DRL tasks. To solve this problem, the source and target domains, source and target tasks of transfer learning are as follows:

Source domain: $\boldsymbol{s}$; target domain: $\boldsymbol{s} \times \boldsymbol{a}$.

Source task: TSI; target task: r, p.

Both source domain and task are subsets of the target domain and task. Therefore, there is a remarkable similarity between the learning process of the source and target tasks, which can be transferred by sharing certain model parameters.

Model-based transfer learning combines samples of the source and target domain to adjust the model parameters. In this study, we use a fine-tuning method to freeze partial layers of the pretrained model and train the remaining layer because samples of source and target domains are the same. Fig. 4 shows that A1 and A2 represent the partial layers of the frozen pretrained model, and B represents the layer to be trained. Based on the fine-tuning method, the neural network of TSA with partial layers frozen is used as the initial network of DRL, and the DRL is retrained.

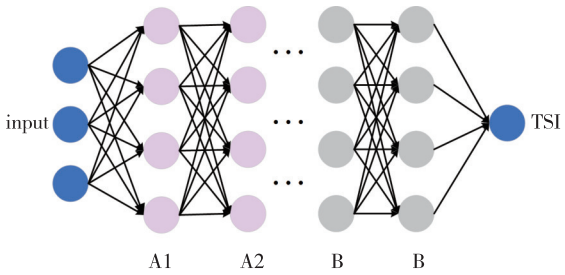

Fig. 4. Schematic diagram of transfer learning.

D. Computational Process

Based on the methods above, the overall block diagram of the proposed TSPC method is shown in Fig. 5. The upper part of the figure is the training process of TSA. In an episode of DRL, the TSA is executed only once. Input to this TSA is the current power flow state and fault location with the most severe instability. Trajectory sensitivity of the generator to TSI and transfer information of DRL are formed based on information of the trained TSA. The lower part is the iterative process of this method. The left is the TSPC process, and the right is the learning process of TD3. Based on the trained TSA of the upper part, actionable devices and transfer parameters are provided to the TD3. The $N-1$ calculation results provide TD3 with current power flow states and rewards, and TD3 gives TSPC process actions to adjust the output-power of generators. After performing a DRL action, the $N-1$ calculation is executed, and the adjustment amount of the actionable generator pair is obtained according to (20). Subsequently, the output-power of generators is adjusted, and the power flow state is updated. Next, the subsequent DRL action is performed. Therefore, the action frequency of DRL is the dispatch frequency of the generators. After several iterations of interaction between the TSPC process and TD3, TD3 can learn a power flow control strategy that satisfies the transient stability constraint eventually.

IV. Case Study

A. Experiment setup

The proposed method was tested on the New England 39-bus standard system [35] and an actual power grid in a region of China; the results are presented in this section. The numerical tests were run on a high-performance computer. TensorFlow was used to write the code in Python (an open-source package).

We randomly selected k generators, changed the output power between 0.8 and 1.6 times, and varied relevant loads and capacitors/reactors to produce 6,561 sets of power-flow data based on the baseline power flow of the New England 39-bus standard system. For each power-flow level, a set of $N-1$ three-phase short-circuit faults is assumed to occur. Simulation time for each scenario was 5 s. Consequently, 6,561 samples were created under these circumstances, of which 5,000 samples were training sets and 1,000 samples were test sets. In the training sets, 2,649 samples are stable and 2,351 are unstable. In the test sets, 531 samples are stable and 469 are unstable. In this study, stratified sampling [36] was used to sample the samples, and results are given in Fig. 6. Fig. 6 shows the active power of the different generators in each sample, with one point in the figure representing one sample. Changes in different generators were considered different layers, and x samples were extracted from each layer. From the figure, it can be seen that active power of different generators in each sample is evenly distributed. The degree of influence of each generator's output on system stability under this distribution is not disturbed by the sample, which enables the neural network to fit the influence of each generator on

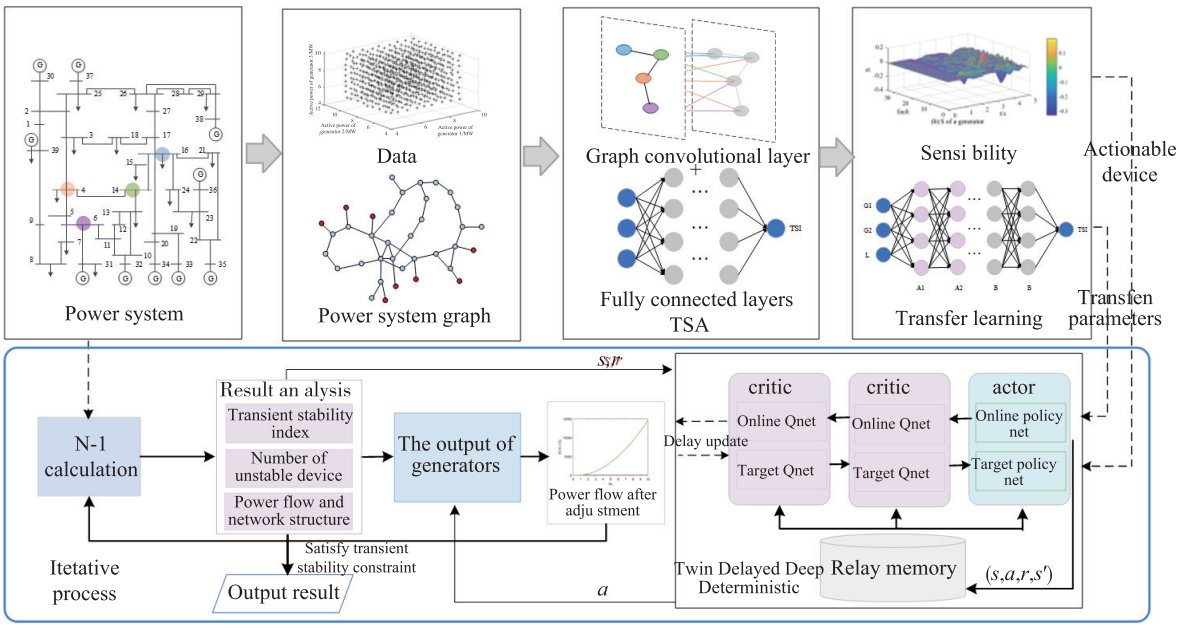

Fig. 5. Overall block diagram of the proposed TSPC method.

system stability accurately and illustrates the reasonableness of the sample distribution.

Figure 7 shows one of the selected samples. When TSI > 0, sample is stable; when $TSI < 0$, sample is unstable. Moreover, the neural network structures are shown in Fig. 8, where (a) and (b) are the neural network structures for DL-based TSA and DRL-based TSPC, respectively. "$(nG+2)\times 1$, $(nG+2)\times 128$, ReLU" refers to the input feature size is $(nG+2)\times 1$, the output feature size is $(nG+2)\times 128$ and activation function is ReLU. Network structure changes with the change of input parameters. As can be seen from the figure, the network structure of DL-based TSA is almost identical to that of DRL-based TSPC, both containing two graph convolution layers and three fully connected layers. The number of fully connected layers here is only the initial number of layers, and the decision of the final optimal number of layers and the number of transferred layers is obtained from Table III.

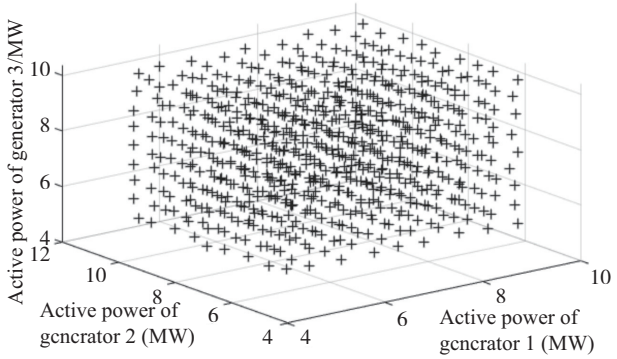

Fig. 6. Stratified sampling result of samples.

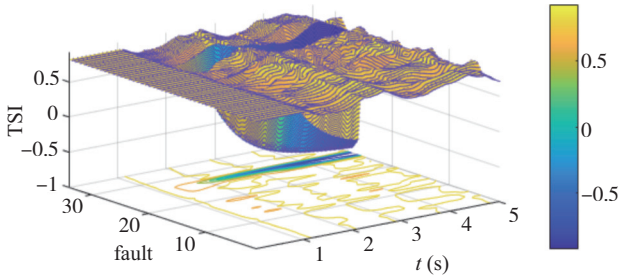

Fig. 7. The parameters of the selected sample.

This study adopted the Heilongjiang power grid in China for the actual power grid. Data generation and distribution methods were consistent with the preceding. The $N-1$ calculation includes faults of 220 kV and higher-voltage lines because of the enormous scale and readily off-limit lines in this scenario. For each contingency, simulation time was 10 s. Parameters of TD3 for different cases are provided in Table I.

TABLE I
PARAMETERS OF TD3 FOR DIFFERENT CASES

System	Learning rate	Batch size	Discount factor	Training time (h)
39-bus system	0.01	100	0.9	20.2
Actual power grid	0.01	200	0.9	55.6

By considering the capacity limits of each generator, the rated capacity of the generator with a smaller capacity limit is used as the capacity limit of the generator pair when acting the actionable generator pair, as shown in (17). If the capacity after action exceeds the capacity limit, the next generator pair can be acted in (17). For the actual power grid, the generator is directly started and stopped according to the actual preventive-control rules, so there is no problem of capacity limitation in the adjustment process.

B. Experimental Results

1) IEEE 39-Bus System

Figure 9 depicts the moving rewards of several approaches

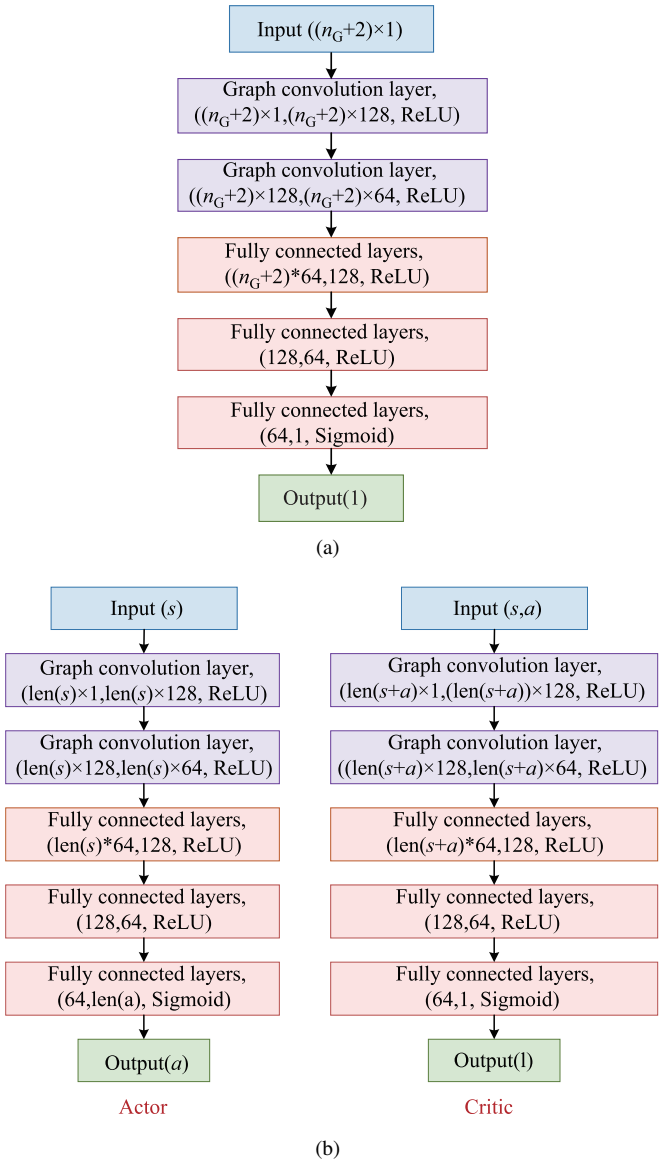

Fig. 8. Neural network structures for DL-based TSA and DRL-based TSPC. (a) Neural network structure for DL-based TSA. (b) Neural network structure for DRL-based TSPC.

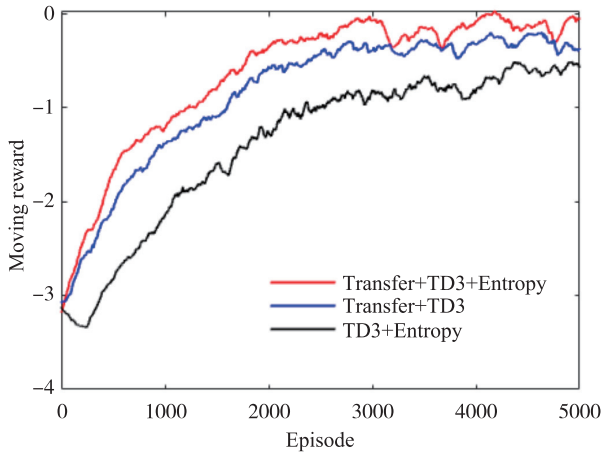

Fig. 9. The moving rewards of different methods in the training process.

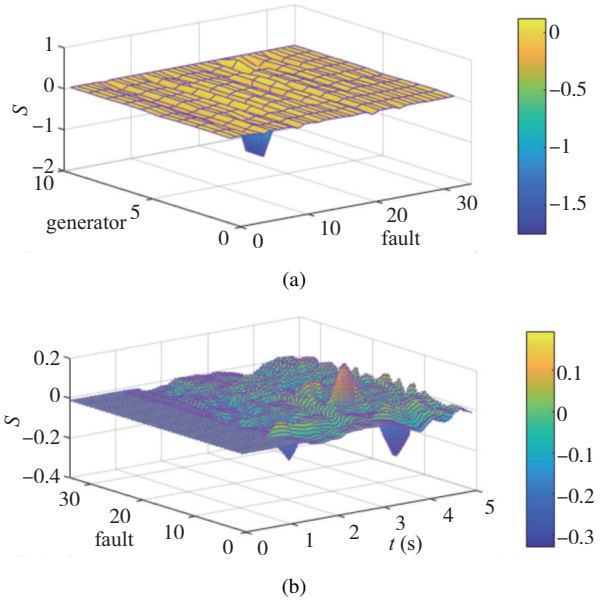

Fig. 10. The sensitivity of active power of the generator to TSI. (a) Sensitivity of active power of each generator to TSI at a particular time; (b) Sensitivity of active power of a generator to TSI at each simulation time.

used in the training process. Curves are the results with transfer learning + TD3 + Entropy, transfer learning + TD3, TD3 + Entropy, respectively. Evidently, compared with other methods, the moving reward of the transfer learning + TD3 + Entropy proposed in this study was higher and converged earlier. This indicates the learning effect is better, and the learning process is faster compared to others. The learning process was accelerated to a certain extent, and the training time and difficulty were reduced owing to transfer learning. Furthermore, adding entropy to TD3 increased the unpredictability of the policy, enabling better exploration and faster learning.

According to Section II-B, sensitivity of the output power of each generator to TSI can be calculated by trained TSA. As shown in Fig. 10, (a) is the sensitivity of the active power of each generator to TSI at a particular time, and (b) is the sensitivity of active power of a generator to TSI at each simulation time. For each fault, there are generators with different sensitivities. As can be seen from the figure, different generators have different sensitivities for different faults, some greater than 0 and some less than 0. In Fig. 10(a), the negative sensitivity of a generator under fault 10 is large, and this generator can then be used as an actionable generator. In Fig. 10(b), at 0–1 s, the system is not faulted and sensitivity is 0. At 3–5 s, sensitivity varies greatly, indicating that when intercepting the sensitivity at a particular moment, sensitivity at a particular moment in 3–5 s is taken to be the best. When $S > 0$, increasing active power of the generator can reduce the degree of instability; when $S < 0$, increasing active power of the generator can increase the degree of instability. According to (17)–(18) and the sensibilities shown in Fig. 10, generator pairs with high sensitivities can be formed and acted, which illustrates the feasibility of the sensitivity calculated by the TSA.

Based on sensitivity calculated based on the TSA as shown

in Fig. 10, actionable generator pairs are formed by the ordering of (17)–(18). Table II shows the actionable generator pairs and corresponding sensibilities under a line fault for IEEE 39-bus system. As can be seen from the table, the highest-priority actionable generator pair includes positively adjusted generators with highest sensitivity and negatively adjusted generators with lowest sensitivity. For the IEEE 39-bus system, there are 10 generators, one of which acts as a slack bus, thus creating a total of 4 pairs of actionable generator pairs. According to (20), if capacity limit is not exceeded, the DRL action maps directly to the highest-priority generator pair, reducing the action space of DRL from 9 generators to 2 generators. Moreover, reduction of action space makes action more targeted and avoids security problems that may occur during power flow adjustment.

TABLE II
ACTIONABLE GENERATOR PAIRS UNDER A LINE FAULT FOR IEEE 39-BUS SYSTEM

Priority	Actionable generator pairs	Sensibilities
1	(7, 5)	(0.011877, −0.017680)
2	(6, 1)	(0.011410, −0.000192)
3 4	(2, 8) (9, 4)	(0.004798, 0.000934) (0.004397, 0.001926)

Table III shows results of different fully connected layers and transfer layers. "2 FCL + 1 Transfer" means there are two FCL (fully connected layers) in the TSA and one transfer layer for DRL. Constraint satisfaction ratio (CSR) refers to the proportion of samples that satisfies transient-stability constraints. The CSR and average reward are the best options when the number of fully connected layers is four and the number of transfer layers is two, as shown in Table III. Thus, this scheme is selected.

TABLE III
TEST RESULTS OF DIFFERENT FULL CONNECTION LAYERS AND DIFFERENT TRANSFER LAYERS

Method	CSR	Average reward
2FCL+1Transfer	97.2%	0.55
3FCL+1Transfer	97.8%	0.59
3FCL+2Transfer	97.6%	0.57
4FCL+2Transfer	98.1%	0.62
4FCL+3Transfer	97.6%	0.56

Figure 11 shows the average adjustment steps (AAS) and CSR with and without transfer learning during iteration for the 39-bus standard system. AAS in Fig. 11 is the rounded value of the average adjustment step in a batch of computations. The average adjustment steps to fulfill the transient stability constraint steadily decrease over the iteration process and tend to stabilize at 9–10 steps after 2500 iterations. During this period, the fraction of samples that fulfill the constraint grows and gradually converges after 2000 iterations. Finally, the samples that fulfill the constraint approached 96%, demonstrating the suggested method can successfully control power flow to meet the transient stability constraint for the 39-bus system. In the presence of transfer learning, both AAS and CSR achieved better results in a shorter number of episodes. After 1500 iterations, the percentage of samples that fulfill the restriction

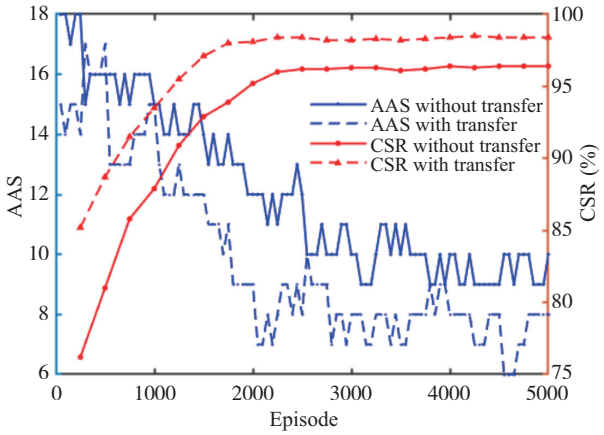

Fig. 11. AAS, CSR with and without transfer learning during iteration for 39-bus standard system.

can exceed 98%, suggesting that transfer learning can speed up learning and enhance its effectiveness.

In Fig. 12, (a) and (b) show the rotor-angle results of transient-stability calculation under faults 1 and 2 before and after control, respectively. The slack bus was designated as the reference, and the rotor angle of other generators was the rotor-angle difference relative to the reference generator. Evidently, under fault 1, one generator of the system was unstable; under fault 2, three generators of the system were unstable. Whether for fault 1 or fault 2, the rotor angle of the generator can be stabilized by this method. Table IV shows the generators under fault 1 before and after control. It can be seen from the figure that active power of each generator is more balanced after control, which indicates the balanced power distribution of the

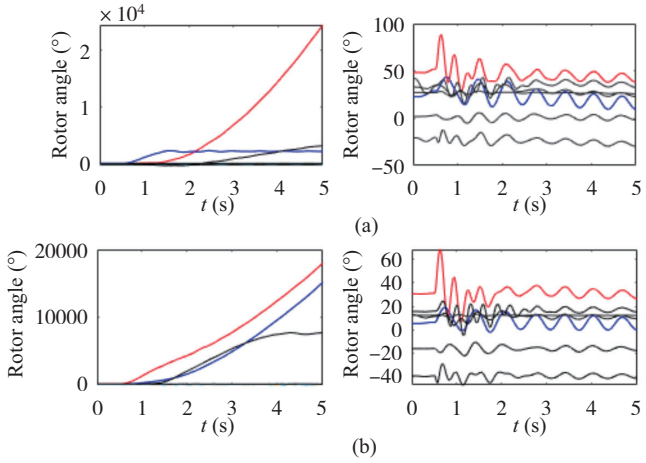

Fig. 12. Rotor angle after transient stability calculation under faults 1 and 2 before and after control for 39-bus standard system. (a) Transient stability calculation under fault 1 before and after control. (b) Transient stability calculation under fault 2 before and after control.

TABLE IV
GENERATORS UNDER FAULT 1 BEFORE AND AFTER CONTROL FOR 39-BUS STANDARD SYSTEM

	Active power output (p.u.)
Generators before control	2.5, 3.13, 2.5, 5.32, 4.08, 4.5, 4.6, 3.4, 2.3, 3.0
Generators after control	2.5, 3.13, 3.5, 3.32, 4.08, 4.5, 4.6, 3.4, 3.3, 3.0

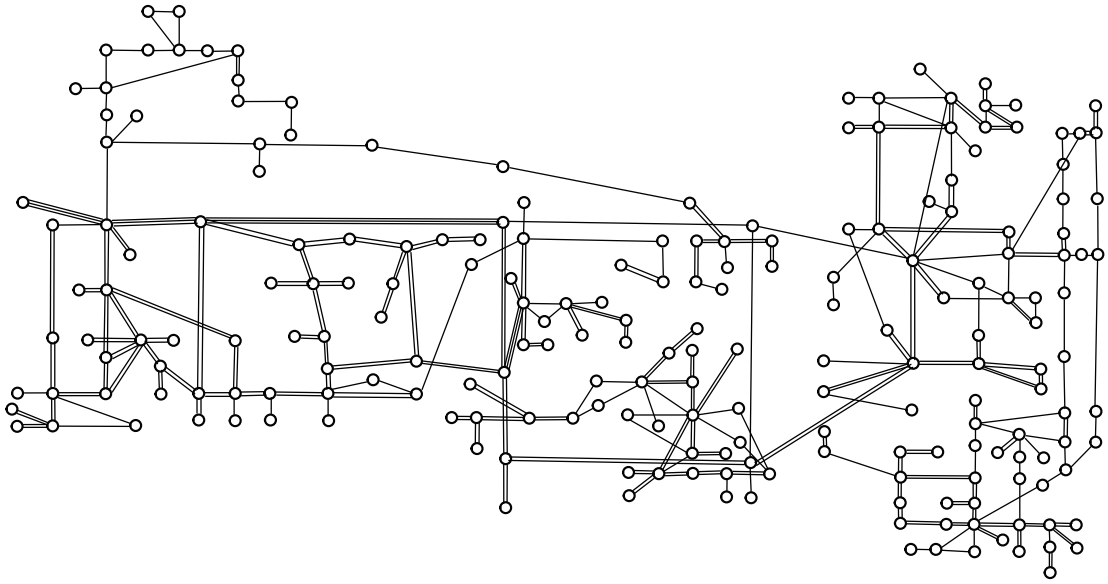

Fig. 13. One-line diagram of Heilongjiang power grid in China.

whole network is conducive to system stability. In addition, a more balanced power flow distribution can also ensure power flow security.

2) Actual Power Grid

The actual power grid has 507 buses, 127 generators, 203 loads, 150 shunt capacitors and reactors, and 436 lines, as depicted in Fig. 13.

Figure 14 illustrates AAS and CSR with and without transfer learning during the iteration phase for the actual power grid. In the process of iteration, AAS gradually decreased from not satisfying to satisfying constraint and was stable at 33 after 8,000 iterations. CSR increased over time, similar to the 39-bus standard system, and eventually converged after 8,000 iterations. Finally, the percentage of samples satisfying the requirement was approximately 92%. The percentage of samples that satisfied the constraint could reach 95% using transfer learning. The AAS increased as compared to the 39-bus standard system, whereas the number of samples that satisfied the limit decreased. However, it also has the effect of controlling power flow such that most samples could satisfy the constraint in limited steps.

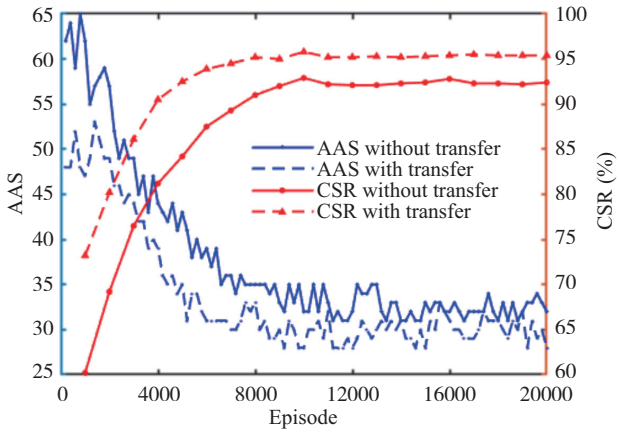

Fig. 14. AAS, CSR with and without transfer learning during iteration process for the actual power grid.

In Fig. 15, (a) and (b) show the rotor-angle results of transient stability calculation under fault 1 and 2 before and after control, respectively. Under fault 1, one generator of the system was unstable; under fault 2, all generators in the observed region were unstable. No matter for fault 1 or fault 2, the rotor angle can be stabilized after the control of this method. Fault 2 is the incoming line fault of the DC-converter station. After its disconnection, all generators in the observed region were unstable. Large-scale transfer of DC-power flow has a significant impact on the system because the incoming line fault of the DC-converter station is equivalent to disconnection of DC. This leads to instability of the generator cluster. Finally, power flow could be controlled to make the system stable, showing he effectiveness of the method under serious fault.

Table V shows generators of the actual power grid before

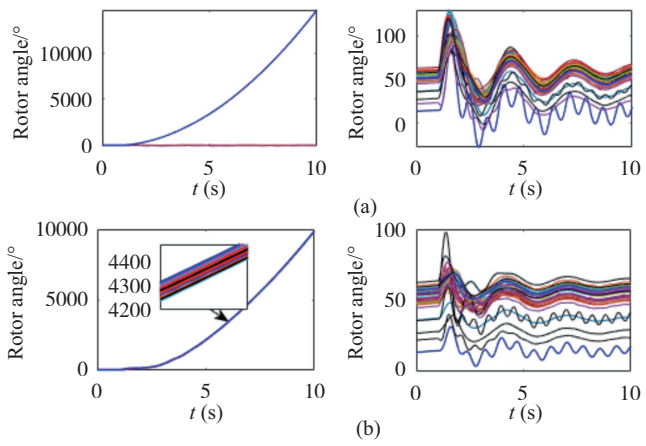

Fig. 15. Rotor angle after transient stability calculation under fault 1 and 2 before and after control for the actual power grid. (a) Transient stability calculation under fault 1 before and after control. (b) Transient stability calculation under fault 2 before and after control.

TABLE V
GENERATORS UNDER FAULT 1 BEFORE AND AFTER CONTROL FOR ACTUAL POWER GRID

	Active power output (p.u.)
Generators before control	0.99, 1.0, 2.0, 2.0, 2.0, 2.0, 2.0, 3.0, 3.0, 3.0, 2.0, 6.0, 3.0, 3.0, 6.0, 1.0, 1.25, 1.25, 3.0, 3.0, 0.99, 0.99, 0.99, 2.15, 2.15, 3.0, 3.5, 6.0, 3.0, 0.99, 2.0, 2.0, 3.5, 3.3, 3.5
Generators after control	0.99, 1.0, 2.0, 2.0, 2.0, 2.0, 2.0, 3.0, 3.0, 3.0, 2.0, 6.0, 3.0, 3.0, 6.0, 1.0, 0.06, 0.99, 1.25, 3.0, 3.0, 0.99, 0.99, 1.375, 0.986, 0.99, 2.15, 3.0, 3.5, 6.0, 3.0, 0.99, 2.0, 0.99, 3.5, 3.3, 3.5, 0.5, 0.5, 0.435, 0.495

and after control under fault 1. As can be seen from the figure, the number of generators increases and the active power of the generators decreases after control, indicating that power distribution of the system is more balanced after control than before. Therefore, we can draw the same conclusions as for the 39-bus standard system.

C. Comparison and Analysis

The proposed method is compared to five algorithms, including "TD3 + Entropy", "Sensibility + TD3", "TSCOPF + Sensibility", "TSCOPF", and SCR, using data from an actual power system. Test results are presented in Table VI. The "TD3 + Entropy" denotes control using only TD3 with Entropy, without TSA and transfer learning. "Sensibility + TD3" is control with TD3 and selection of action targets using trajectory sensitivity. "TSCOPF + ANN" and "TSCOPF" are solving this problem using TSCOPF with and without TSA based on the artificial neural network (ANN). The "TSCOPF + ANN" method refers to Reference [37] to use ANN to predict the rotor-angle transient stability margin and to be incorporated into the TSCOPF problem. The "TSCOPF" leverages particle swarm optimization to solve the TSCOPF problem [38]. SCR reschedules power generation with trajectory sensitivity to ensure system stability [39]. The unit control cost is 1$/MW.

TABLE VI
TEST RESULTS OF OTHER METHODS

Method	CSR	Control cost ($)	Control time (s)
This method	94.6%	7.8	208
TD3 + Entropy	80.5%	12.3	672
Sensibility + TD3	90.2%	10.9	458
TSCOPF + ANN [37]	92.5%	8.2	241
TSCOPF [38]	88.6%	8.8	372
SCR [39]	90.9%	20.6	777

Table VI shows that the effect is poor with "TD3 + Entropy". Since DRL is built without any action space constraint, action space is vast, and learning efficiency is insufficient for a large-scale power grid. It also demonstrates that sensitivity calculated by TSA can reduce action space and improve learning efficiency. "Sensibility + TD3" has good effect. Action space is restricted based on trajectory sensitivity of actionable devices, resulting in excellent action efficiency. However, without transfer learning, learning efficiency decreases. For the "TSCOPF + ANN", the effect is slightly worse compared to the proposed method, and 7.5% of samples do not satisfy the constraint. Because transient-stability constraints are simplified with TSA based on ANN, and the calculation effect is improved. Nevertheless, samples that satisfy the requirements become less owing to poor convergence of TSCOPF in the actual large-scale power grid. With TSCOPF, the effect is poor because of the complexity of transient-stability constraints and poor convergence, and only 88.6% of samples satisfy the constraint. For SCR, although CSR is good, its control cost and adjustment time are extremely high. This approach cannot learn the best strategy because it uses trial and error to adjust, leading to long control time and high control cost. In terms of control time and cost, the proposed method has more advantages compared to other methods because it considers the control cost in reward and can be trained offline and tested online. In summary, the proposed method is better than other methods in both control effect and time.

Figure 16 shows the ratio of acted generators for TSPC when the TSA is composed of GCNN, CNN and ANN. The least number of acted generators is found under GCNN for both the 39-bus standard system and the actual power grid, indicating that sensitivity calculated by GCNN guides the adjustment of generators more accurately, and the transfer of the parameters learned by GCNN to DRL is more conducive to the learning and action of DRL than by other neural networks.

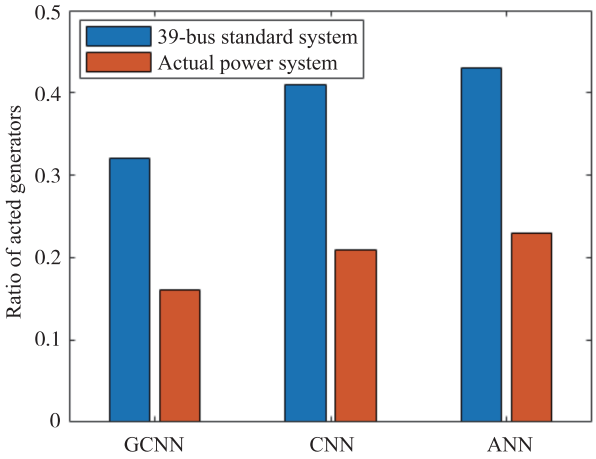

Fig. 16. Ratio of acted generators for TSPC when the TSA is composed of GCNN, CNN and ANN.

V. CONCLUSION

This study proposed a TSPC method by combining TSA based on GCNN with transfer DRL. Parameters learned by the TSA can be used to approximate the sensitivity of each generator to the TSI and can transfer the learned knowledge to DRL to improve learning efficiency. A standard system and an actual power grid are utilized to verify the method.

The proposed method effectively controlled power flow to satisfy transient-stability constraints and achieved automated control of the power-flow state for 39-bus standard system and actual power grid. Moreover, transfer learning can accelerate learning and improve effectiveness of learning. Control effect of the proposed method was better than of previous methods, and the control time was faster. Generators with considerable influence are identified by calculating the sensitivity of each generator to TSI through parameters of trained TSA, and

action space is narrowed. Moreover, based on transfer learning, knowledge learned by trained neural networks is transferred to DRL, exploiting TSA information, reducing training time, and improving learning efficiency.

Rotor angle instability only is considered in this study. In future studies, rotor angle and transient voltage instability will be considered simultaneously, and a TSPC satisfying these two instability constraints will be constructed. Additionally, from the control range perspective, stability of one region is currently controlled. In the future, stability of multiple regions will be considered to realize coordinated control between different regions.

REFERENCES

[1] S. Chu and A. Majumdar, "Opportunities and challenges for a sustainable energy future," *Nature*, vol. 488, no. 7411, pp. 294–303, Aug. 2012.

[2] S. Liu, Y. Li, X. Liu, T. Zhao and P. Wang, "Resilient Power Systems Operation with Offshore Wind Farms and Cloud Data Centers," *in CSEE Journal of Power and Energy Systems*, vol. 9, no. 6, pp. 1985–1998, Nov 2023.

[3] T. Su, Y. B. Liu, J. B. Zhao, and J. Y. Liu, "Deep belief network enabled surrogate modeling for fast preventive control of power system transient stability," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 315–326, Jan. 2022.

[4] Q. Y. Jiang and G. C. Geng, "A reduced-space interior point method for transient stability constrained optimal power flow," *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1232–1240, Aug. 2010.

[5] G. C. Geng and Q. Y. Jiang, "A two-level parallel decomposition approach for transient stability constrained optimal power flow," *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 2063–2073, Nov. 2012.

[6] Y. Yuan, J. Kubokawa, and H. Sasaki, "A solution of optimal power flow with multicontingency transient stability constraints," *IEEE Transactions on Power Systems*, vol. 18, no. 3, pp. 1094–1102, Aug. 2003.

[7] P. L. C. Wieler, R. Kuiava, and W. F. S. Souza, "Transient stability constrained optimal power flow based on trajectory sensitivity for power dispatch of distributed synchronous generators," *IEEE Latin America Transactions*, vol. 18, no. 7, pp. 1247–1254, Jul. 2020.

[8] A. Pizano-Martínez, C. R. Fuerte-Esquivel, E. A. Zamora-Cárdenas, and J. M. Lozano-García, "Directional derivative-based transient stability-constrained optimal power flow," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3415–3426, Sep. 2017.

[9] S. Batchu and K. Teeparthi, "A preventive transient stability control strategy through individual machine equal area criterion framework," *IEEE Access*, vol. 9, pp. 167776–167794, Dec. 2021.

[10] S. W. Xia, Z. H. Ding, M. Shahidehpour, K. W. Chan, S. Q. Bu, and G. Y. Li, "Transient stability-constrained optimal power flow calculation with extremely unstable conditions using energy sensitivity method," *IEEE Transactions on Power Systems*, vol. 36, no. 1, pp. 355–365, Jan. 2021.

[11] H. Saberi, T. Amraee, C. Zhang, and Z. Y. Dong, "A heuristic benders-decomposition-based algorithm for transient stability constrained optimal power flow," *Electric Power Systems Research*, vol. 185, pp. 106380, Aug. 2020.

[12] H. Bouchekara, "Solution of the optimal power flow problem considering security constraints using an improved chaotic electromagnetic field optimization algorithm," *Neural Computing and Applications*, vol. 32, no. 7, pp. 2683–2703, Apr. 2020.

[13] H. L. Yuan and Y. Xu, "Trajectory sensitivity based preventive transient stability control of power systems against wind power variation," *International Journal of Electrical Power & Energy Systems*, vol. 117, pp. 105713, May 2020.

[14] H. L. Yuan and Y. Xu, "Preventive-corrective coordinated transient stability dispatch of power systems with uncertain wind power," *IEEE Transactions on Power Systems*, vol. 35 no. 5, pp. 3616–3626, Sep. 2020.

[15] S. Ghosh and B. H. Chowdhury, "Security-constrained optimal rescheduling of real power using Hopfield neural network," *IEEE Transactions on Power Systems*, vol. 11, no. 4, pp. 1743–1748, Nov. 1996.

[16] T. J. Liu, Y. B. Liu, J. Y. Liu, L. F. Wang, L. X. Xu, G. Qiu, and H. J. Gao, "A bayesian learning based scheme for online dynamic security assessment and preventive control," *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 4088–4099, Sep. 2020.

[17] S. T. Zhang, D. X. Zhang, J. Qiao, X. Y. Wang, and Z. J. Zhang, "Preventive control for power system transient security based on XGBoost and DCOPF with consideration of model interpretability," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 279–294, Mar. 2021.

[18] F. Tian, X. X. Zhou, Z. H. Yu, D. Y. Shi, Y. Chen, and Y. H. Huang, "A preventive transient stability control method based on support vector machine," *Electric Power Systems Research*, vol. 170, pp. 286–293, May 2019.

[19] Q. Y. Li, Y. Q. Yu, T. Lin, X. Y. Fu, H. Du, and X. L. Xu, "Deep reinforcement learning-based fast prediction of strategies for security control," in *Proceedings of 2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2)*, 2021, pp. 2737–2742.

[20] T. J. Wang and Y. Tang, "Parallel deep reinforcement learning-based power flow state adjustment considering static stability constraint," *IET Generation, Transmission & Distribution*, vol. 14, no. 25, pp. 6276–6284, Dec. 2020.

[21] H. Zeng, Y. Zhou, Q. Guo, Z. Cai and H. Sun, "Distributed Deep Reinforcement Learning-based Approach for Fast Preventive Control Considering Transient Stability Constraints," *CSEE Journal of Power and Energy Systems*, vol. 9, no. 1, pp. 197–208, Jan. 2023.

[22] Z. M. Zhang, R. Yao, S. W. Huang, Y. Chen, S. W. Mei, and K. Sun, "An online search method for representative risky fault chains based on reinforcement learning and knowledge transfer," *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 1856–1867, May 2020.

[23] H. Yin, Z. H. Ou, J. J. Fu, Y. F. Cai, S. Chen, and A. B. Meng, "A novel transfer learning approach for wind power prediction based on a serio-parallel deep learning architecture," *Energy*, vol. 234, pp. 121271, Nov. 2021.

[24] C. Ren and Y. Xu, "Transfer learning-based power system online dynamic security assessment: using one model to assess many unlearned faults," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 821–824, Jan. 2020.

[25] M. He, J. S. Zhang, and V. Vittal, "Robust online dynamic security assessment using adaptive ensemble decision-tree learning," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4089–4098, Nov. 2013.

[26] R. Ramirez, Y. C. Chiu, S. Y. Zhang, J. Ramirez, Y. D. Chen, Y. F. Huang, and Y. F. Jin, "Prediction and interpretation of cancer survival using graph convolution neural networks," *Methods*, vol. 192, pp. 120–130, Aug. 2021.

[27] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[28] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 3844–3852.

[29] R. S. Sutton and A. G. Barto, *Reinforcement Learning*: *An Introduction*, 2nd ed., Cambridge: MIT Press, 2018.

[30] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the International Conference on Machine Learning (PMLR)*, 2018, pp. 1582–1591.

[31] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine. (2019, Jan. 29). Soft actor-critic algorithms and applications. arXiv: 1812.05905. [Online]. Available: https://arxiv.org/pdf/1812.05905.pdf.

[32] Q. Shi, H. K. Lam, C. B. Xuan, and M. Chen, "Adaptive neuro-fuzzy PID controller based on twin delayed deep deterministic policy gradient algorithm," *Neurocomputing*, vol. 402, pp. 183–194, Aug. 2020.

[33] L. Pan, Q. P. Cai, and L. B. Huang, "Softmax deep double deterministic policy gradients," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020, pp. 987.

[34] S. T. Niu, Y. X. Liu, J. Wang, and H. B. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 151–166, Oct. 2020.

[35] Information Trust Institute. (2018, Dec. 17). IEEE 39-Bus test case archive. [Online]. Available: https://icseg.iti.illinois.edu/ieee-39-bus-system/.

[36] K. Lang, E. Liberty, and K. Shmakov, "Stratified sampling meets machine learning," in *Proceedings of the 33rd International Conference on Machine Learning (PMLR)*, 2016, pp. 2320–2329.

[37] R. A. Lajimi and T. Amraee, "A two stage model for rotor angle transient stability constrained optimal power flow," *International Journal of Electrical Power & Energy Systems*, vol. 76, pp. 82–89, Mar. 2016.

[38] N. Mo, Z. Y. Zou, K. W. Chan, and T. Y. G. Pong, "Transient stability constrained optimal power flow using particle swarm optimisation," *IET Generation, Transmission & Distribution*, vol. 1, no. 3, pp. 476–483, May 2007.

[39] T. B. Nguyen and M. A. Pai, "Dynamic security-constrained rescheduling of power systems using trajectory sensitivities," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 848–854, May 2003.

Tianjing Wang received a Ph.D. degree from China Electric Power Research Institute, Beijing, China, in 2022. She is currently a Research Fellow at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. From Aug. 2021 to Feb. 2022, she was a visiting scholar at the school of Electrical and Electronic Engineering, Nanyang Technological University. Her research interests include deep reinforcement learning applications in power system control, transfer learning and federated learning applications in power systems.

Yong Tang received a Ph.D. degree in Power System and Automation from China Electric Power Research Institute, Beijing, China, in 2002. He is currently a Chair Professor with CEPRI. He is the Fellow of CSEE and Senior Member of IEEE. His research interests are power system simulation and analysis, voltage stability and control, load modeling, and simulation.