

Intelligent Predetermination of Generator Tripping Scheme: Knowledge Fusion-based Deep Reinforcement Learning Framework

Lingkang Zeng, Wei Yao, *Senior Member, IEEE*, Ze Hu, Hang Shuai, *Member, IEEE*, Zhouping Li, Jinyu Wen, *Member, IEEE*, and Shijie Cheng, *Life Fellow, IEEE, Fellow, CSEE*

Abstract—Generator tripping scheme (GTS) is the most commonly used scheme to prevent power systems from losing safety and stability. Usually, GTS is composed of offline predetermination and real-time scenario match. However, it is extremely time-consuming and labor-intensive for manual predetermination for a large-scale modern power system. To improve efficiency of predetermination, this paper proposes a framework of knowledge fusion-based deep reinforcement learning (KF-DRL) for intelligent predetermination of GTS. First, the Markov Decision Process (MDP) for GTS problem is formulated based on transient instability events. Then, linear action space is developed to reduce dimensionality of action space for multiple controllable generators. Especially, KF-DRL leverages domain knowledge about GTS to mask invalid actions during the decision-making process. This can enhance the efficiency and learning process. Moreover, the graph convolutional network (GCN) is introduced to the policy network for enhanced learning ability. Numerical simulation results obtained on New England power system demonstrate superiority of the proposed KF-DRL framework for GTS over the purely data-driven DRL method.

Index Terms—Deep reinforcement learning, generator tripping scheme, graph convolutional network, invalid action masking, knowledge fusion.

I. INTRODUCTION

TRANSIENT stability emergency control (TSEC), or system integrity protection schemes (SIPS) are the essential measures to prevent safety and stability loss from contingencies and faults [1], [2]. So far, there have been three main strategies of TSEC: (i) offline predetermination and real-time scenario match [3], [4]; (ii) online predetermination and real-time scenario match [5], [6] and (iii) real-time control [7],

[8]. Due to high requirements for the regularly super-real-time parallel simulation [9], strategy (ii) and (iii) have not been widely adopted in practical grids yet. On the contrary, offline predetermination does not take up online response time and has been commonly used all over the world [10].

Figure 1 shows the framework of the strategy (i). Offline predetermination is designed to formulate effective TSEC in advance for those contingencies that would result in transient instability under the planned operating conditions [11]. The effectiveness of the formulated TSEC needs to be verified through transient simulation [12]. Then, all these predetermined TSEC together with corresponding instability scenarios are saved in the *Look-up Table*. During online operation, once a severe disturbance occurs, real-time scenario match will be conducted immediately to find the appropriate TSEC in the *Look-up Table* [13]. The real-time match is finished so quickly the TSEC can be implemented as soon as possible to prevent transient instability. Here, this paper only focuses on offline predetermination of TSEC, hereinafter referred to as predetermination.

Among the various SIPS summarized from global industry experiences and academic research [14], [15], generator tripping scheme (GTS) is the most effective and commonly used scheme against transient rotor angle instability (TRAI) [13], [16]. Currently, predetermination of GTS is mainly accomplished manually by experts in practice [11], [17], [18]. Obviously, manually offline predetermination will be extremely time-consuming and labor-intensive in large-scale modern power systems, where there exist lots of typical operating conditions and conceivable contingencies. In recent years, some optimal methods are proposed to auto-search for effective and economical control measures [19], [20]. However, search space will be very large when there are lots of controllable generators.

Fortunately, application of deep reinforcement learning (DRL) techniques in power system have attracted more and more attention in recent years [21]–[25]. It provides a new feasible way for power system stability control, for example, autonomous voltage control [26], [27], automatic generation control [28], preventive control [29], and load frequency control [30]. When it comes to predetermination of GTS, the core and essential requirement is that a well-trained DRL agent can formulate effective GTS for the transient instability scenario through as few attempts as possible. However, direct

Manuscript received December 25, 2022; revised March 28, 2023; accepted May 19, 2023. Date of online publication December 28, 2023; date of current version January 5, 2024. This work was supported by National Natural Science Foundation of China (No. U22B20111, No. U1866602).

L. K. Zeng, W. Yao (corresponding author, email: w.yao@hust.edu.cn), Z. Hu, Z. P. Li, J. Y. Wen, and S. J. Cheng are with State Key Laboratory of Advanced Electromagnetic Engineering and Technology, School of Electrical and Electronics Engineering, Huazhong University of Science and Technology, Wuhan 430074, China. L. K. Zeng is also with Dispatching and Control Center, Central China Branch of State Grid Corporation of China, Wuhan 430077, China.

H. Shuai is with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996, USA.

DOI: 10.17775/CSEEJPES.2022.08970

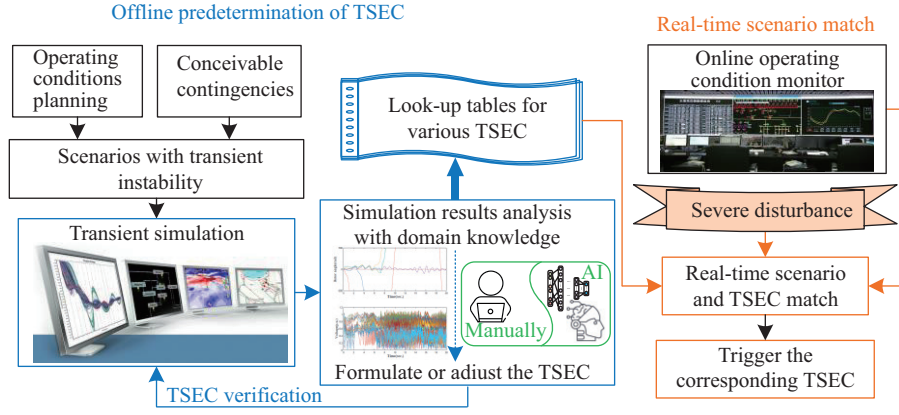


Fig. 1. The commonly used TSEC strategy: offline predetermination and real-time scenario match.

application of the aforementioned DRL methods for GTS faces the following problems.

First, the MDP of response based TSEC are formulated along a time series within a single simulation [31], [32]. Whereas, as a kind of event-triggered one-shot control, GTS needs to be adjusted and predetermined through multiple repetitive transient simulations. Simulations are repeatedly conducted for the same scenario in Fig. 1 with different GTSs until an effective GTS is obtained. Thus MDP for the predetermination of GTS should be reformulated. Second, action space increases exponentially with increase of controllable generators. To this end, a multi-agent DRL (MA-DRL) is proposed to reduce action space of each agent [30], [32] and improve scalability [33]. The attention mechanism is further introduced into MA-DRL to reduce agent training time [34]. Whereas, application and development of MA-DRL still has many limitations, such as difficulty of model training increases sharply due to centralized learning of multi-agents [21]. Third, exploitation of domain knowledge and special properties are supposed to improve training efficiency and guarantee performance [35]. The knowledge-assist learning proposed in [36] introduced another predictor network, which increases training difficulty. GTS related knowledge fusion in [37] provides the agent with a guide to make decisions. Nevertheless, constraints for useless or bad actions are ignored there, which have significant impact on training efficiency. Overall, there has been rare direct knowledge exploitation in decision-making yet among existing DRL based TSEC research.

Considering the above application challenges, this paper proposes a knowledge fusion-based deep reinforcement learning (KF-DRL) framework for intelligent predetermination of GTS. The proposed KF-DRL can directly exploit domain knowledge during decision-making and adjust GTS during the MDP based on transient simulation. Numerous simulations are conducted on the New England power system under different scenarios. Results verify the well-trained KF-DRL can efficiently predetermine GTS for never seen scenarios. The main contributions of this work are summarized here:

- A framework of KF-DRL is proposed for offline predetermination of GTS. It enables transformation of predetermination of GTS from manual style to intelligent style through the combination of domain knowledge and

DRL algorithm.

- A novel MDP formulation method for GTS is developed based on transient instability events, which enables adjustment of GTS according to simulation verification results. In addition, high dimensionality is avoided by the linear action space design.
- To directly leverage domain knowledge during the decision-making process, a novel knowledge fusion style is proposed in the manner of invalid action masking. It significantly improves performance of making effective GTS for never seen scenarios.
- To enhance learning ability and speed up convergence rate, the proposed KF-DRL adopts a graph convolutional network (GCN) embedding to additionally extract topological features of power systems.

The rest of this paper is organized as follows. Section II introduces MDP for the predetermination of GTS and linear action space design. Section III presents the framework of KF-DRL and implementation manner of knowledge fusion. Numerous simulation and result analysis are given in Section IV. Section V concludes this paper.

II. GENERATOR TRIPPING SCHEME PREDETERMINATION

A. GTS Predetermination as a Markov Decision Process

Figure 2 presents the framework of GTS predetermination based on double deep-Q-network (DDQN) [38]. At the beginning of a MDP, transient simulation results of the instability scenario without GTS are observed as the initial state s_1 . Then, DDQN based agent outputs action a_1 according to the policy network and formulates the corresponding measures. Transient simulation is conducted again to verify effectiveness of the latest measures and the observed state will be updated. States observed by the agent vary from action to action. In this way, the agent can adjust actions according to updated states. Transient simulations and action adjustment will be conducted in turns until the effective GTS is obtained or the number of adjustments reaches the upper limit.

Concretely, MDP consists of state s_k , action a_k , reward r_k , terminal T_k^{done} and state transition function as follows:

1) *State*: The state s_k refers to observed transient simulation results at decision step k . The heterogeneous state can be described as:

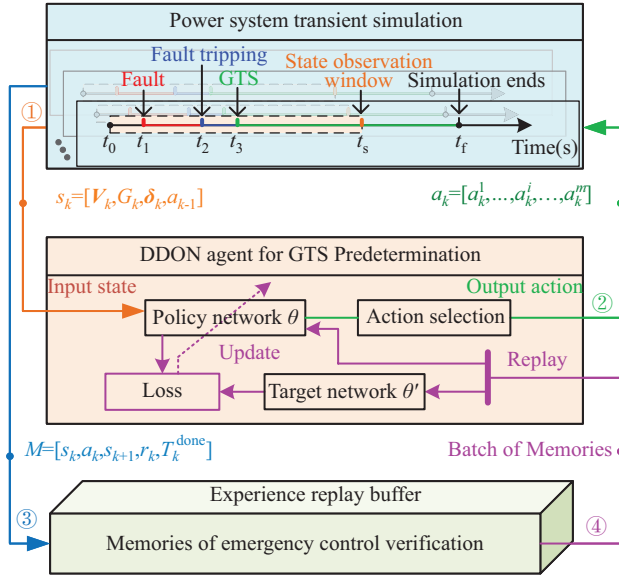


Fig. 2. The framework of the DDQN based GTS predetermination.

$$s_k = [V_k, G_k, \delta_k, a_{k-1}] \quad (1)$$

where V_k represents observed voltage magnitudes of buses at decision step k . G_k refers to topology of the power system after tripping out the fault line, which is represented as a graph. δ_k represents rotor angles of generators relative to the reference generator. a_{k-1} represents the GTS taken in the verification simulation at decision step $k-1$.

2) *Action*: The action a_k refers to the tripping amount of generators at decision step k , which can be described as:

$$a_k = [a_{k1}^1, \dots, a_{k1}^i, \dots, a_{k1}^m] \quad (2)$$

where m is the number of the controllable generators. a_{k1}^i represents the tripping amount of generator i at decision step k . Here, assume each controllable generator refers to a power plant which includes of ten identical small-capacity generators. The value of a_{k1}^i can be from 0 to 1.0 with interval of 0.1.

In order to balance exploitation and exploration of experience during the training process, ϵ -greedy strategy is utilized for action selection. It is formulated as:

$$a_k^* = \arg \max_{a_k} Q(s_k, a_k) \quad (3)$$

$$P(a_k = a_k^*) = 1 - \epsilon_k \quad (4)$$

$$\epsilon_k = \min\{\epsilon_0 - n_s \Delta \epsilon, \epsilon_f\} \quad (5)$$

where a_k^* is action with the biggest Q value evaluated by policy network. $P(a_k = a_k^*)$ is probability of selecting a_k^* at decision step k , while the greedy coefficient ϵ_k represents the probability of selecting action randomly. ϵ_0 and ϵ_f represent initial and final value of greedy coefficient, respectively. n_s is number of training steps and $\Delta \epsilon$ is attenuation of the ϵ .

3) *State transition function*: The state transition of the MDP is realized by transient simulation. As shown in Fig. 2, simulation begins at $t_0 = 0$ s and ends at t_f . During the simulation, the contingency or severe fault occurs at t_1 , which is followed by tripping off fault line at t_2 . Then, GTS are taken to controlled generators at t_3 . The state observation window is

set as $[t_0, t_s]$, where $t_3 \leq t_s \leq t_f$. It is transient data between $[t_3, t_s]$ that varies from action to action, which makes the agent adjust actions according to different states.

4) *Terminal*: T_k^{done} is a variable in bool type, which represents whether MDP terminates. Concretely, T_k^{done} is *True* when determined GTS can maintain transient stability, otherwise *False*. It can be formulated as:

$$T_k^{\text{done}} = \begin{cases} 0, & \exists \delta_{k,t_f}^{ij} \geq 360^\circ \cup k > N_L \\ 1, & \forall \delta_{k,t_f}^{ij} < 360^\circ \cap k \leq N_L \end{cases} \quad (6)$$

where δ_{k,t_f}^{ij} represents relative power angle between generator i and j at simulation time t_f . N_L is limit of iteration number for predetermination of one GTS.

5) *Reward*: The reward $r_k(s_k, a_k)$, referred to r_k , depends on state transition after verification of action a_k by transient simulation. r_k is formulated as:

$$\begin{cases} r_k = -r_s \Delta A_k^s + r_d T_k^{\text{done}} \\ \Delta A_k^s = \sum_{i=1}^m a_{k1}^i - \sum_{i=1}^m a_{k-1}^i \end{cases} \quad (7)$$

where r_s is the penal coefficient for power loss resulting from generator tripping. ΔA_k^s is increment of tripping amount. r_d is reward for GTS that can maintain transient stability against contingencies. The reward is designed to decrease tripping amount as much as possible in the promise of maintaining transient stability.

The elements of MDP are obtained during interaction between the agent and transient simulation, and will be saved as a memory into the replay buffer. The memories in the buffer will be sampled regularly by experience replay to update weights of the policy network. As shown in Fig. 2, memory is formulated as:

$$M = [s_k, a_k, s_{k+1}, r_k, T_k^{\text{done}}] \quad (8)$$

B. Linear Action Space for Multi-generator Power Systems

Generally, all possible actions should be included in the action space of the DRL based agent. Each action will be represented by an individual neuron, which is the well-known *one-hot* coding. However, assuming there are m controllable generators in the power system, exponential action space design is presented in Fig. 3. There are h tripping choices for each generator. Then, number of actions in the exponential action space is h^m . It is obvious the difficulty of model training and consumption of computation resource will increase rapidly with increase of controllable generators m .

To overcome this problem, linear action space is introduced into the DDQN based agent for GTS [39]. As shown in Fig. 3, the number of the neurons for linear action space is $h \cdot m$. Action selection in the linear action space will be as:

$$a_k^{x*} = \arg \max_{a_k^x} Q(s_k, a_k^x) \quad (9)$$

$$P(a_k^x = a_k^{x*}) = 1 - \epsilon_k \quad (10)$$

$$\begin{cases} x = h(i-1) + z, & x \in \{1, 2, \dots, h \cdot m\} \\ i \in \{1, 2, \dots, m\}, & z \in \{1, 2, \dots, h\} \end{cases} \quad (11)$$

where a_k^{x*} is action with the biggest Q value among the $h \cdot m$ actions. x represents z -th tripping choice of generator i . Hence,

a_k^x can be also regarded as tripping action of generator i at decision step k , which is a_k^i . All the left elements except for a_{k-1}^i in the last GTS a_{k-1} will keep unchanged after decision step k . In this situation, each element of a_k in (2), referred to a_k^j , is obtained as:

$$a_k^j = \begin{cases} a_k^x, & j = i \\ a_{k-1}^j, & j \neq i \end{cases} \quad (12)$$

As shown in Fig. 3, for the exponential action space, it theoretically only takes one decision step, at least, to obtain the effective GTS. For the linear action space, it would be K^* decision steps, at least, where $1 \leq K^* \leq m$. Specifically, theoretical minimum K^* depends on how many generators are needed to be involved in the GTS. Essentially, linear action space sacrifices the quite limited decision steps in exchange for lower difficulty of agent training, which is absolutely cost-effective and necessary for power systems with multiple controllable generators.

III. KNOWLEDGE FUSION BASED DRL FOR GTS

A. Framework and Implementation of KF-DRL

Waste of domain knowledge about GTS would result in the unnecessary explorations and slow convergency of the training process. In order to obtain better performance, framework of the knowledge fusion is proposed in Fig. 4 for the DDQN based agent, which can exploit domain knowledge during action selection.

As shown in Fig. 4, knowledge fusion is a kind of constrain for action selection, which is updated after formulation of a_k and plays a role at the next decision step for a_{k+1} . Specifically, knowledge fusion is implemented in the manner of invalid action masking [40]. Among the complete action space, the invalid actions are identified by analysis of transient simulation results using related domain knowledge. Then, invalid action list will be updated and actions on this list will be masked and definitely not selected for the next action a_{k+1} .

In this paper, two knowledge rules are applied to the DDQN based agent for predetermination of GTS. They are repeated action masking and bad action masking. In the actual power grid, decision space of the agent would be pruned through knowledge, and then the agent independently would explore in the remaining space. A balance between knowledge fusion and autonomous exploration could be realized.

1) *Repeated action masking*: It is found DDQN based agents tend to give the same action decisions for different state inputs during training. A number of action adjustments will be wasted for repeated invalid decisions. As a result, the likelihood of successfully formulating an effective GTS within limited adjustment number will be reduced.

To overcome this problem, repeated action masking is introduced to the DDQN based agent. As shown in Fig. 4, there is a list L_k to save historical actions formulated by agent before MDP ends.

$$L_k = \{a_1, \dots, a_\tau, \dots, a_k\}, \quad 1 \leq \tau \leq k \quad (13)$$

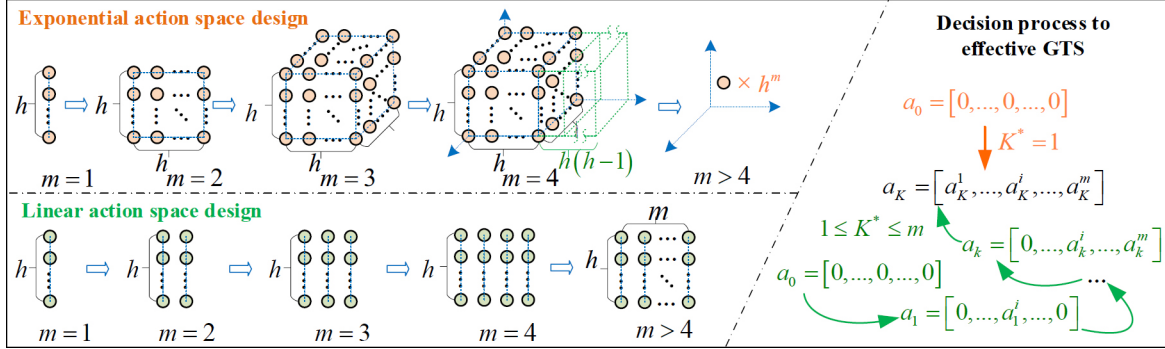


Fig. 3. The action space design of the DDQN based agent for GTS predetermination.

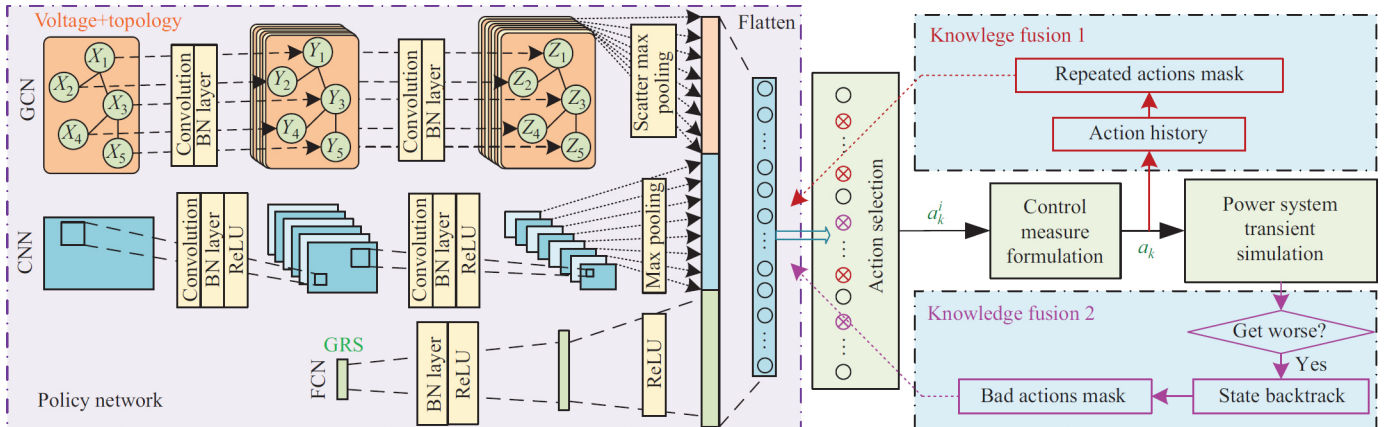


Fig. 4. The framework of knowledge fusion based DDQN agent for GTS predetermination.

Every time action a_k is formulated, collection of repeated actions C_{k+1}^{rep} will be updated as:

$$C_{k+1}^{\text{rep}} = \{a_{k+1}^x | a_{k+1} = a_\tau\} \quad (14)$$

where a_τ represents every action in historical actions list L_k . That means C_{k+1}^{rep} is the collection of actions a_{k+1}^x that make a_{k+1} be the same as any one action in L_k . It is obvious C_{k+1}^{rep} contains different action choices for each controllable generator.

2) *Bad action masking*: The performance of transient stability will get worse, if improper control measures are taken. For example, tripping of non-critical generators or over-tripping of critical generators could result in worse transient performance. Motivated by this, the second knowledge fusion is formulated according to result analysis of verification simulation for a_k .

For transient performance assessment, critical instability time (CIT) of verification simulations for a_{k-1} and a_k will be compared. The comparison result determines whether bad action masking will be applied to action selection for a_{k+1} . CIT for a_k is defined as:

$$CIT(a_k) = \min\{t | \max \Delta \delta_{k,t} \geq 360^\circ\} \quad (15)$$

where t represents sampling time when maximum relative power angle is over 360° during verification simulation for a_k . $CIT(a_k)$ is minimum of t in the collection. Before the MDP ends, if $CIT(a_k) \leq CIT(a_{k-1})$, bad action masking will be applied to the action selection for a_{k+1} . Otherwise, the a_k^x decided at step k is beneficial to transient stability and the bad action masking will not be formulated for a_{k+1} .

For state backtrack, since a_k^x decided at step k resulted in worse transient performance, state obtained after verification of a_k should be erased in MDP. In this way, bad action a_k^x will not influence follow-up decisions of MDP. Specifically, state and action should backtrack one step.

For masking formulation, collection of bad actions C_{k+1}^{bad} will be formulated according to (10) as:

$$C_{k+1}^{\text{bad}} = \{a_{k+1}^y | x \leq y \leq h * i\} \quad (16)$$

where x comes from bad action a_k^x . The collection C_{k+1}^{bad} includes not only a_{k+1}^x , but also actions whose tripping amounts are larger than a_{k+1}^x for generator i .

With the effect of the invalid action masking, action selection (10) can be modified as:

$$P(a_k^x = a_k^{x*}) = 1 - \varepsilon_k, \quad a_k^x \notin (C_k^{\text{rep}} \cup C_k^{\text{bad}}) \quad (17)$$

Whether during training or evaluation, each decision updates the set of action masking to guide the next decision. The difference is that, during the training process, as performance of the agent increases, repeated and bad actions will gradually decrease, and probability of triggering action masking will decrease. During the evaluation process, the agent has been trained well, repeated and bad actions will basically not appear, and probability of triggering action blocking is very low.

B. Policy Network Design and Input Pre-processing

Figure 4 shows architecture of the designed policy network.

It contains three different networks to extract features from the heterogeneous state formulated in (1). The graph convolutional network (GCN) is adopted for feature extraction of bus voltages V_k and topology G_k of power systems. The convolutional neural network (CNN) is used for feature extraction of rotor angles δ_k of generators. The feature of the action verified at last decision step a_{k-1} is extracted by the fully connected network (FCN). Then, the features extracted by different networks will be spliced and fed to the output layer.

Pre-processing for V_k , G_k and δ_k is as:

1) *Voltages*: The voltage magnitude of buses, V_k , can be represented as:

$$V_k = \begin{bmatrix} V_k^{1,1} & \dots & V_k^{1,\Omega} \\ \vdots & \ddots & \vdots \\ V_k^{N,1} & \dots & V_k^{N,\Omega} \end{bmatrix} \quad (18)$$

where $V_k^{N,\Omega}$ represents voltage magnitude of bus N at sampling point Ω during verification simulation. N is total number of the buses. Ω is last sampling point of the observation window shown in Fig. 2, where $\Omega = f_s(t_3 - t_0)$. f_s is the sampling frequency.

2) *Topology*: The structure of power system is a typical graph, where buses and transmission lines are vertices and edges of the graph, respectively. Graph G_k at decision step k can be represent as:

$$G_k = (B_k, T_k) \quad (19)$$

where $B_k = \{B_k^1, B_k^2, \dots, B_k^N\}$ and $T_k = \{T_k^1, T_k^2, \dots, T_k^M\}$ represent buses and transmission lines, respectively. M refers to number of lines of the power system. If there is a transmission line connecting bus B_k^i and B_k^j , there is $T_k^p = (B_k^i, B_k^j)$ in T_k . Then the adjacency matrix of the power system is defined as $G_k \in \mathbb{R}^{N \times N}$, which is also a part of the input state in (1). Element g_k^{ij} can be represented as:

$$g_k^{ij} = \begin{cases} 1, & (B_k^i, B_k^j) \in T_k \\ 0, & (B_k^i, B_k^j) \notin T_k \end{cases} \quad (20)$$

3) *Rotor angle*: The rotor angles of generators, δ_k , can be obtained as follow:

$$\delta_k = \begin{bmatrix} \delta_k^{1,0} & \dots & \delta_k^{1,\Omega} \\ \vdots & \ddots & \vdots \\ \delta_k^{m,1} & \dots & \delta_k^{m,\Omega} \end{bmatrix} \quad (21)$$

where $\delta_k^{m,\Omega}$ represents rotor angle of generator m relative to reference generator at sampling point Ω . δ_k needs to be pre-processed.

$$\delta_k^{i,j} \leftarrow \tanh(\delta_k^{i,j}/2\pi) \quad (22)$$

C. Training Process of the Proposed KF-DRL Algorithm

The training objective of the KF-DRL based agent is to minimize temporal difference (TD) error by updating weights of the policy network, which is formulated as:

$$e(\theta_j) = \sum_{i=1}^{\Phi} \gamma Q'(s_{k+1}, \arg \max_a Q(s_{k+1}, a | \theta_j) | \theta_j')$$

$$+ r_k(1 - T_k^{\text{done}}) - Q(s_k, a_k | \theta_j) \quad (23)$$

where Q and Q' represent the policy network and target network, respectively. θ_j and θ'_j represent weight parameter of Q and Q' at j -th weights update. Φ represents batch size of experience replay. γ is discount factor. $e(\theta_j)$ represents batch TD errors between Q values evaluated by policy network and target Q values at the j -th weight update. Accordingly, weight update is conducted as:

$$\theta_{j+1} = \theta_j - \alpha \nabla_{\theta} e(\theta_j) \quad (24)$$

where α is learning rate.

The pseudo-code of the proposed KF-DRL algorithm is presented in Algorithm 1.

Algorithm 1: Training process of the KF-DRL for offline predetermination of GTS

- 1 Set total training episodes N_T , memory capacity D , minibatch Φ , synchronization frequency f_s , discount factor γ , learning rate α , greedy coefficient ε , iteration limit N_L .
 - 2 Initialize the networks θ, θ' , training step $n_s = 0$.
 - 3 **while** $n_s \leq N_T$ **do**
 - 4 Select an operating condition and a contingency
 - 5 Observe s_1 ▷ (1), (18)–(21)
 - 6 Update greedy coefficient ε ▷ (5)
 - 7 **while** $T_k^{\text{done}} = \text{False}$ or $k \leq N_L$ **do**
 - 8 Select an action a_k^x ▷ (17)
 - 9 Formulate control measure a_k ▷ (11)–(12)
 - 10 Verify a_k through transient simulation
 - 11 Observe $s_{k+1}, r_k, T_k^{\text{done}}, CIT(a_k)$ ▷ (1), (6)–(7), (15)
 - 12 **if** $CIT(a_k) \leq CIT(a_{k-1})$ **then**
 - 13 State backtrack and formulate C_{k+1}^{bad} ▷ (16)
 - 14 **end**
 - 15 Formulate C_{k+1}^{rep} ▷ (13)–(14)
 - 16 Save memory into the replay buffer ▷ (8)
 - 17 **end**
 - 18 Sample minibatch of memories $\Phi \sim D$
 - 19 Update policy network θ ▷ (23)–(24)
 - 20 **if** $n_s \equiv 0 \bmod f_s$ **then**
 - 21 Update target network $\theta' \leftarrow \theta$
 - 22 **end**
 - 23 $n_s \leftarrow n_s + 1$
 - 24 **end**
 - 25 Return the well-trained policy network
-

IV. CASE STUDY

A. Test System and Parameters Design

Numerous verification simulations are carried out on the IEEE New England power system shown in Fig. 5. During simulation, time parameters are set as: $t_1 = 1$ s, $t_3 = 1.3$ s, $t_s = 6$ s, $t_f = 10$ s. The sampling frequency is $f_s = 20$ Hz. Note $G10$ is selected as reference generator in this paper and will not be considered for GTS. The fault duration is set to 0.2–0.25 s to generate transient power angle instability samples to

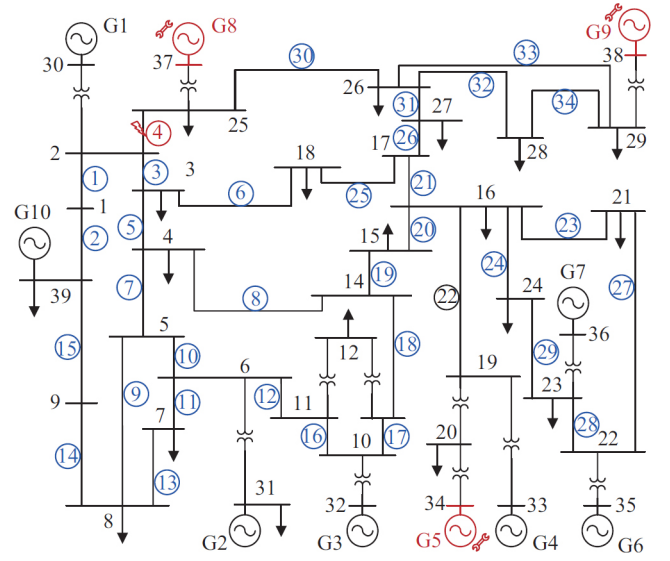


Fig. 5. IEEE New England power system.

verify effectiveness of the proposed strategy. A total of 595 TRAI scenarios are obtained in simulation conditions shown in Table I. Then, 100 scenarios are randomly selected as testing scenarios.

TABLE I
SETTINGS OF THE SIMULATION SCENARIOS

Conditions	Settings	Number
Power and load levels	[1.0, 1.05], [1.1, 1.15], [1.2, 1.25]	3
Fault lines	34 lines except for ②	33
Fault location	10%, 20%, ..., 90%	9
Fault duration ($t_2 - t_1$)	0.2, 0.25	2

The policy network of the KF-DRL based agent is designed as follows: two convolution layers and one pooling layer are designed for both GCN and CNN network. The number of output neurons of all three channels is set as 100. The choice of tripping ratio for each generator is set as $[0, 0.1, \dots, 1.0]$, which is 11 in total. Therefore, dimensionality of linear action space is set as 99. Hyper-parameters are set as follows: $N_T = 30000$, $D = 10000$, $\Phi = 128$, $f_s = 20$, $\gamma = 0.1$, $\alpha = 0.0001$, $\varepsilon_0 = 0.9$, $\Delta\varepsilon = 0.0001$, $\varepsilon_f = 0.05$, $N_L = 50$, $r_s = 1$, and $r_d = 10$.

Table II shows the different methods for comparison. M5 is the KF-DRL based method proposed in this paper. Note M3 adopts CNN to extract features of the bus voltage. M1 is a commonly used pure data-driven agent. M4 is a commonly used CNN-based agent. All transient simulation and network training are conducted on a computer with Intel Core (TM)

TABLE II
METHODS FOR COMPARISON

Method	Details			
	Masking C_k^{rep}	Masking C_k^{bad}	Voltages	Topology
M1	×	×	✓	✓
M2	✓	×	✓	✓
M3	✓	✓	×	×
M4	✓	✓	✓	×
M5	✓	✓	✓	✓

i9-10900K @ 3.70 GHz CPU, 64 GB RAM. The simulation model is built in the Power System Analysis Software Package. The training code is written in Python with PyTorch.

B. Effectiveness Verification of the Proposed KF-DRL

The proposed KF-DRL based agent for offline predetermination of GTS is trained on training scenarios. During the whole training process, performance of the policy network is evaluated on testing scenarios every 2000 episodes. The performance comparison of the 15 evaluations are given in Fig. 6.

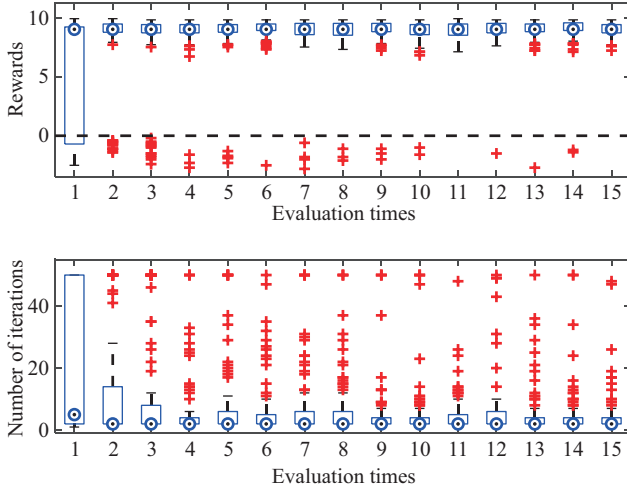


Fig. 6. The performance evaluated on the testing scenarios during the training of the proposed KF-DRL based agent.

Each reward here represents accumulated rewards of all actions determined for one scenario. The negative value of reward means failure the agent cannot give effective GTS within limited iterations. The number of iterations refers to the length of MDP for one scenario. As shown in Fig. 6, overall distribution of rewards continues to increase with the training process. The number of scenarios with negative rewards decreases. Especially, it becomes 0 at final evaluation, which means the effective GTSs are obtained for all testing scenarios. This result verified effectiveness of the proposed KF-DRL method for GTS predetermination.

Figure 7 shows the decision process for a testing scenario at final evaluation. Three actions determined by the proposed KF-DRL are a_1^{90} , a_2^{79} and a_3^{46} , which mean tripping 10% power of G9, tripping 10% power of G8 and tripping 10% power of G5, respectively. Finally, effective actions can be represented as $a_3 = [0, 0, 0, 0, 0.1, 0, 0, 0.1, 0.1]$.

C. Domain Knowledge Fusion for GTS

In order to demonstrate effects of knowledge fusion, performance of M1, M2 and M5 (the proposed KF-DRL method) are evaluated on testing scenarios every 200 training episodes. The training processes of all the three methods are repeated 5 times, and results comparison is shown in Fig. 8. Concretely, total rewards, number of iterations, and number of successfully formulating GTS have been counted and compared. Note higher rewards means less amount of generator tripping.

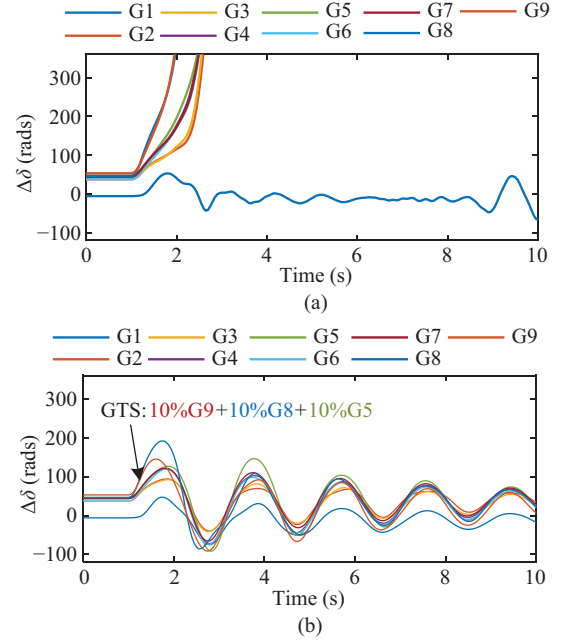


Fig. 7. The transient simulation verification of the control measures formulated by the proposed method. (a) Rotor angles of generators without any control. (b) Rotor angles of generators with the third control.

The fewer number of iterations means higher efficiency of predetermination.

For M1 and M2, repeated action masking saves number of iterations for exploration of other potentially effective actions. As a result, the number of testing scenarios where the agent successfully formulates effective GTS increased significantly. For M2 and M5, bad action masking further improved quality of memories at the beginning of the training process. Hence, convergence rate of the proposed KF-DRL method is faster than M2.

D. Feature Extraction Using GCN

To demonstrate benefits of using GCN for topological feature extraction, performance of M3, M4, and M5 are evaluated during the training process. Comparison results are presented in Fig. 9. Thanks to GCN, the feature of bus voltage can be extracted together with topological information. That means the policy network of the proposed method has stronger representation learning ability than M4 using CNN. Therefore, convergence rate of the proposed KF-DRL method is faster than M4, and predetermination performance is slightly better.

V. CONCLUSION

This paper has proposed a framework of knowledge fusion-based deep reinforcement learning for offline intelligent predetermination of power system GTS. The MDP was first formulated based on transient simulation. In addition, a linear action space was designed to reduce dimensionality. Then, knowledge fusion was developed for decision-making in the manner of invalid action masking. Besides, GCN was adopted to extract topological features for the sake of enhanced learning ability.

Simulation results in the New England power system verified effectiveness and efficiency of the proposed KF-DRL

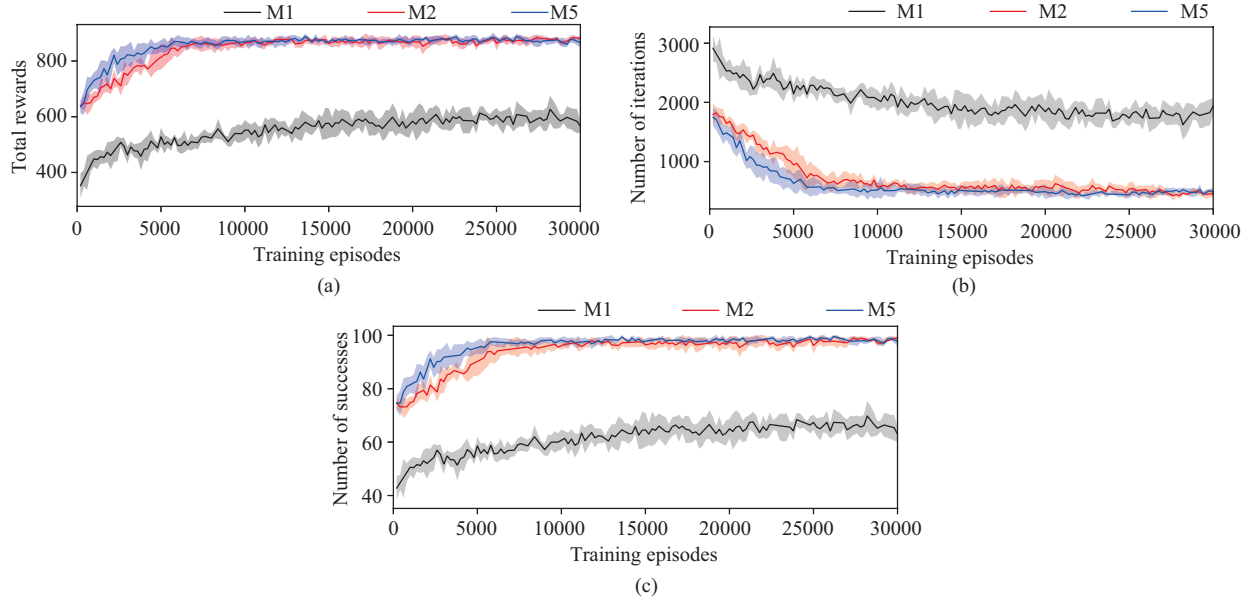


Fig. 8. The evaluated performance of different agents with/without KF (M1: without any knowledge fusion. M2: with the only repeated action masking. M5: the proposed KF-DRL method with the repeated and bad action masking. All the three methods adopt the GCN to extract the topological feature.) (a) Total episode rewards. (b) Total times of iteration. (c) Number of cases with effective measures.

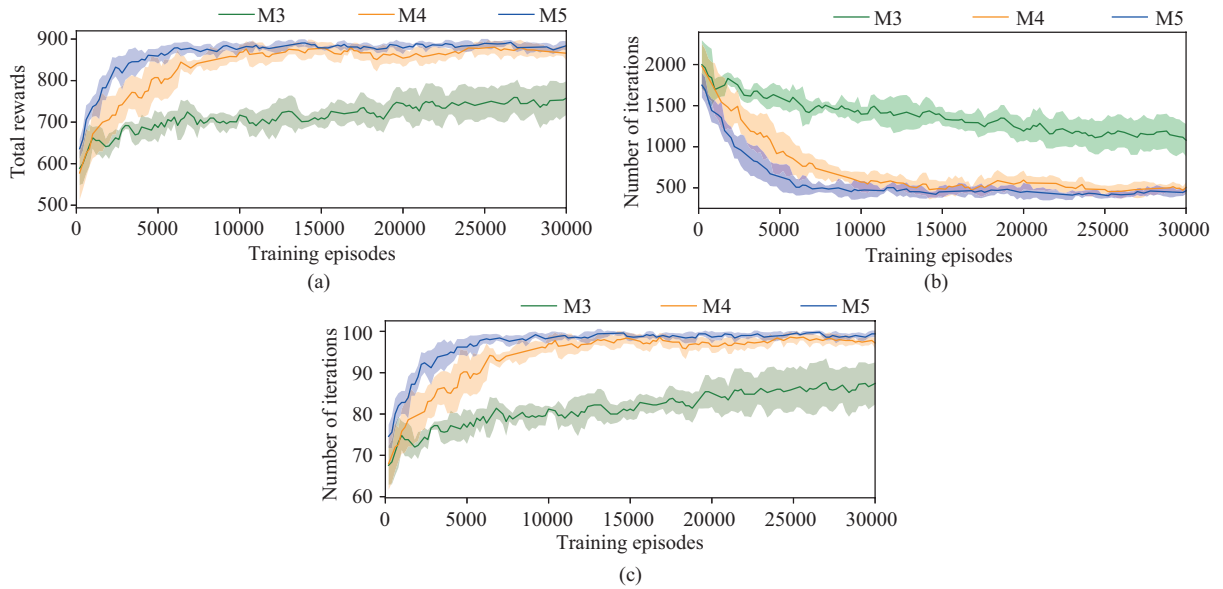


Fig. 9. The evaluated performance of different agents with/without GCN (M3: without extracting the bus voltage and topological features. M4: without topological feature. M5: the proposed KF-DRL method.) (a) Total episode rewards. (b) Total times of iteration. (c) Number of cases with effective measures.

framework. Specifically, knowledge fusion of repeated action masking significantly improved success rate of formulating effective GTS. Furthermore, knowledge fusion of bad action masking can accelerate the convergency rate of the training process. In addition, the usage of GCN for topological feature extraction can bring not only faster convergency rate, but also a slight improvement of performance on predetermination of GTS. The proposed framework can be applied to other control domains. Future work will focus on application in large-scale power grids.

REFERENCES

- [1] J. A. López and C. N. Lu, "Adaptable system integrity protection scheme considering renewable energy sources output variations," *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 3459–3469, Sep. 2020.
- [2] M. S. Ballal, A. R. Kulkarni, and H. M. Suryawanshi, "Methodology for the improvements in synchrophasor based System Integrity Protection Schemes under stressed conditions," *Sustainable Energy, Grids and Networks*, vol. 26, pp. 100465, Jun. 2021.
- [3] P. M. Anderson and B. K. LeVerend, "Industry experience with special protection schemes," *IEEE Transactions on Power Systems*, vol. 11, no. 3, pp. 1166–1179, Aug. 1996.
- [4] B. Sahoo and S. R. Samantaray, "System Integrity Protection Scheme for enhancing backup protection of transmission lines," *IEEE Systems Journal*, vol. 15, no. 3, pp. 4578–4588, Sep. 2021.
- [5] H. Ota, Y. Kitayama, H. Ito, N. Fukushima, K. Omata, K. Morita, and Y. Kokai, "Development of transient stability control system (TSC system) based on on-line stability calculation," *IEEE Transactions on Power*

- Systems*, vol. 11, no. 3, pp. 1463–1472, Aug. 1996.
- [6] X. Xu, H. X. Zhang, C. G. Li, Y. T. Liu, W. Li, and V. Terzija, "Optimization of the event-driven emergency load-shedding considering transient security and stability constraints," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2581–2592, Jul. 2017.
 - [7] F. Tian, X. Zhang, Z. H. Yu, W. J. Qiu, D. Y. Shi, J. Qiu, M. Liu, Y. L. Li, and X. X. Zhou, "Online decision-making and control of power system stability based on super-real-time simulation," *CSEE Journal of Power and Energy Systems*, vol. 2, no. 1, pp. 95–103, Mar. 2016.
 - [8] A. C. Adewole, R. Tzoneva, and A. Apostolov, "Adaptive under-voltage load shedding scheme for large interconnected smart grids based on wide area synchrophasor measurements," *IET Generation Transmission & Distribution*, vol. 10, no. 8, pp. 1957–1968, May 2016.
 - [9] Z. H. Li, G. Q. Yao, G. C. Geng, and Q. Y. Jiang, "An efficient optimal control method for open-loop transient stability emergency control," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2704–2713, Jul. 2017.
 - [10] V. Madani, D. Novosel, S. Horowitz, M. Adamiak, J. Amantegui, D. Karlsson, S. Imai, and A. Apostolov, "IEEE PSRC report on global industry experiences with system integrity protection schemes (SIPS)," *IEEE Transactions on Power Delivery*, vol. 25, no. 4, pp. 2143–2155, Oct. 2010.
 - [11] Z. W. Yao, V. R. Vinnakota, Q. Zhu, C. Nichols, G. Dwernychuk, and T. Inga-Rojas, "Forewarned is forearmed: An automated system for remedial action schemes," *IEEE Power and Energy Magazine*, vol. 12, no. 3, pp. 77–86, May/Jun. 2014.
 - [12] H. Khoshkhoo, S. Yari, A. Pouryekt, V. K. Ramachandaramurthy, and J. M. Guerrero, "A remedial action scheme to prevent mid/long-term voltage instabilities," *IEEE Systems Journal*, vol. 15, no. 1, pp. 923–934, Mar. 2021.
 - [13] D. Min, S. J. Kim, S. Seo, Y. H. Moon, K. Sun, J. H. Chow, and K. Hur, "Computing safety margins of a generation rejection scheme: A framework for online implementation," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 2337–2346, May 2018.
 - [14] N. K. Rajalwal and D. Ghosh, "Recent trends in integrity protection of power system: A literature review," *International Transactions on Electrical Energy Systems*, vol. 30, pp. e12523, May 2020.
 - [15] M. K. Jena, S. R. Samantaray, and B. K. Panigrahi, "A new wide-area backup protection scheme for series-compensated transmission system," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1877–1887, Sep. 2017.
 - [16] D. H. Choi, S. H. Lee, Y. C. Kang, and J. W. Park, "Analysis on special protection scheme of Korea electric power system by fully utilizing STATCOM in a generation side," *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 1882–1890, May 2017.
 - [17] E. Ghahremani, A. Heniche-Oussedik, M. Perron, M. Racine, S. Landry, and H. Akremi, "A detailed presentation of an innovative local and wide-area special protection scheme to avoid voltage collapse: From proof of concept to grid implementation," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5196–5211, Sep. 2019.
 - [18] S. H. Lee, W. C. Sung, J. H. Liu, Y. Y. Hong, Y. H. Hong, C. H. Wang, and C. C. Chu, "Applications of system integrity protection scheme and multi-phase reclosing of transmission lines for stability enhancement in Taiwan Power System," *IEEE Transactions on Industry Applications*, vol. 57, no. 5, pp. 4548–4557, Sep/Oct. 2021.
 - [19] Z. H. Li, G. C. Geng, and Q. Y. Jiang, "Transient stability emergency control using asynchronous parallel mixed-integer pattern search," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2976–2985, Jul. 2018.
 - [20] G. X. Gan, Z. X. Zhu, G. C. Geng, and Q. Y. Jiang, "An efficient parallel sequential approach for transient stability emergency control of large-scale power system," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 5854–5864, Nov. 2018.
 - [21] X. Chen, G. N. Qu, Y. J. Tang, S. Low, and N. Li, "Reinforcement learning for selective key applications in power systems: Recent advances and future challenges," *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 2935–2958, Jul. 2022.
 - [22] C. Huang, H. C. Zhang, L. Wang, X. Luo, and Y. H. Song, "Mixed deep reinforcement learning considering discrete-continuous hybrid action space for smart home energy management," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 3, pp. 743–754, May 2022.
 - [23] J. Zhao, F. X. Li, S. Mukherjee, and C. Sticht, "Deep reinforcement learning-based model-free on-line dynamic multi-microgrid formation to enhance resilience," *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 2557–2567, Jul. 2022.
 - [24] Z. T. Zhang, J. Shi, W. W. Yang, Z. F. Song, Z. X. Chen, and D. Q. Lin, "Deep Reinforcement Learning Based Bi-layer Optimal Scheduling for Microgrids Considering Flexible Load Control," *CSEE Journal of Power and Energy Systems*, vol. 9, no. 3, pp. 949–962, May 2023.
 - [25] S. Y. Chen, J. J. Duan, Y. Y. Bai, J. Zhang, D. Shi, Z. W. Wang, X. Z. Dong, and Y. Z. Sun, "Active Power Correction Strategies Based on Deep Reinforcement Learning—Part II: A Distributed Solution for Adaptability," *CSEE Journal of Power and Energy Systems*, vol. 8, no. 4, pp. 1134–1144, Jul. 2022.
 - [26] S. Y. Wang, J. J. Duan, D. Shi, C. L. Xu, H. F. Li, R. S. Diao, and Z. W. Wang, "A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4644–4654, Nov. 2020.
 - [27] T. J. Wang, Y. Tang, "Automatic Voltage Control of Differential Power Grids Based on Transfer Learning and Deep Reinforcement Learning," *CSEE Journal of Power and Energy Systems*, vol. 9, no. 3, pp. 937–948, May 2023.
 - [28] J. W. Li, T. Yu, and X. S. Zhang, "Emergency fault affected wide-area automatic generation control via large-scale deep reinforcement learning," *Engineering Applications of Artificial Intelligence*, vol. 106, pp. 104500, Nov. 2021.
 - [29] H. T. Zeng, Y. Z. Zhou, Q. L. Guo, Z. M. Cai, and H. B. 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.
 - [30] Z. M. Yan and Y. Xu, "A multi-agent deep reinforcement learning method for cooperative load frequency control of a multi-area power system," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4599–4608, Nov. 2020.
 - [31] Q. H. Huang, R. K. Huang, W. T. Hao, J. Tan, R. Fan, and Z. Y. Huang, "Adaptive power system emergency control using deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1171–1182, Mar. 2020.
 - [32] C. Y. Chen, M. J. Cui, F. X. Li, S. F. Yin, and X. N. Wang, "Model-free emergency frequency control based on reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2336–2346, Apr. 2021.
 - [33] D. Chen, K. A. Chen, Z. J. Li, T. S. Chu, R. Yao, F. Qiu and K. X. Li, "PowerNet: Multi-agent deep reinforcement learning for scalable powergrid control," *IEEE Transactions on Power Systems*, vol. 37, no. 2, pp. 1007–1017, Mar. 2022.
 - [34] F. Yang, D. H. Huang, D. D. Li, S. F. Lin, S. M. Mueyen, and H. B. Zhai "Data-driven load frequency control based on multi-agent reinforcement learning with attention mechanism," *IEEE Transactions on Power Systems*, vol. 38, no. 6, pp. 5560–5569, Nov. 2023, doi: 10.1109/TPWRS.2022.3223255.
 - [35] W. Q. Cui, Y. Jiang, and B. S. Zhang, "Reinforcement learning for optimal primary frequency control: A lyapunov approach," *IEEE Transactions on Power Systems*, vol. 38, no. 2, pp. 1676–1688, Mar. 2023.
 - [36] H. Zhao, J. H. Zhao, J. Qiu, G. Q. Liang, and Z. Y. Dong, "Cooperative wind farm control with deep reinforcement learning and knowledge-assisted learning," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 6912–6921, Nov. 2020.
 - [37] Z. P. Li, L. K. Zeng, W. Yao, Z. Hu, H. Shuai, Y. Tang, and J. Y. Wen "Intelligent emergency generator rejection schemes based on knowledge fusion and deep reinforcement learning," *Proceedings of the CSEE*, to be published.
 - [38] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 2094–2100.
 - [39] A. Kanervisto, C. Scheller, and V. Hautamäki, "Action space shaping in deep reinforcement learning," in *2020 IEEE Conference on Games (CoG)*, 2020, pp. 479–486, doi: 10.1109/CoG47356.2020.9231687.
 - [40] S. Y. Huang and S. Ontañón. (2020, Jun.). A closer look at invalid action masking in policy gradient algorithms. [Online]. Available: <https://www.researchgate.net/publication/342464961>



Linggang Zeng received the B.S. and Ph.D. degrees in Electrical Engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2016 and 2022, respectively. He was a Visiting Student Researcher with the University of Rhode Island, Kingston, RI, USA, from 2019 to 2020. Currently, he is with the Dispatching and Control Center of Central China Branch of State Grid Corporation of China. His research interests include deep reinforcement learning, power system security assessment and stability control.



Wei Yao (M'13–M'17) received the B.S. and Ph.D. degrees in Electrical Engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2004 and 2010, respectively. He was a Postdoctoral Researcher with the Department of Power Engineering, HUST, from 2010 to 2012, and a Postdoctoral Research Associate with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, U.K., from 2012 to 2014. He is currently a Professor with the School of Electrical and Electronics Engineering, HUST.

His current research interests include power system stability analysis and control, renewable energy, HVDC and DC grid, and application of artificial intelligence in smart grid.



Zhouping Li received the B.S. and M.S. degrees in Electrical Engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2019 and 2022, respectively. His research interests include deep reinforcement learning, power system stability control and fault diagnosis.



Ze Hu received the B.S. degree in Electrical Engineering from Shandong University, Jinan, China, in 2017. He is currently working toward the M.S. degree in Electrical Engineering with Huazhong University of Science and Technology (HUST), Wuhan, China. His current research interest focuses on the application of artificial intelligence methods in power system stability control.



Jinyu Wen (M'10) received the B.S. and Ph.D. degrees in Electrical Engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1992 and 1998, respectively. He was a visiting student from 1996 to 1997, a Research Fellow from 2002 to 2003 with the University of Liverpool, Liverpool, U.K., and a Senior Visiting Researcher with the University of Texas at Arlington, Arlington, USA, in 2010. From 1998 to 2002, he was the Director Engineer with XJ Electric Company Ltd., China. In 2003, he joined HUST, where he is

currently a Professor with the School of Electrical and Electronics Engineering. His current research interests include renewable energy integration, energy storage, multi-terminal HVDC, and power system operation and control.



Hang Shuai (S'17–M'19) received the B.Eng. degree from Wuhan Institute of Technology (WIT), Wuhan, China, in 2013, and the Ph.D. degree in Electrical Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2019. He was also a Visiting Student Researcher with the University of Rhode Island (URI), Kingston, RI, USA, from 2018 to 2019. He was a Postdoctoral Researcher with the University of Rhode Island from 2019 to 2020. Currently, he is a Research Scientist with the University of Tennessee,

Knoxville, TN, USA. His research interests include deep reinforcement learning, microgrid optimization, bulk power system resilience.



Shijie Cheng (M'86–SM'87–F'11–LF'18) received the B.S. degree in Electrical Engineering from Xi'an Jiaotong University, Xi'an, China, in 1967, the M.Sc. degree in Electrical Engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1981, and the Ph.D. degree in Electrical Engineering from the University of Calgary, Calgary, AB, Canada, in 1986, respectively. He has been a Professor with the School of Electrical and Electronics Engineering, HUST, Wuhan, China, since 1991. His research interests include power

system control, stability analysis, application of Artificial Intelligence, and energy storage. Prof. Cheng is an Academician of the Chinese Academy of Sciences.