Energy Engineering

Tech Science Press

Doi:10.32604/ee.2025.073912

ARTICLE

Curriculum-Learning-Guided Multi-Agent Deep Reinforcement Learning for N-1 Static Security Prevention and Control

Ximing Zhang[1,*], Zhuohuan Li[2], Xuexia Quan[1], Kai Cheng[2] and Yang Yu[2]

[1]China Southern Power Grid Co., Ltd., Guangzhou, 510700, China
[2]Digital Grid Research Institute Co., Ltd., China Southern Power Grid, Guangzhou, 510663, China
*Corresponding Author: Ximing Zhang. Email: zhangxm@csg.cn or ximing1980@yeah.net

Received: 28 September 2025; Accepted: 21 November 2025; Published: 06 August 2026

ABSTRACT: The "N-1" criterion represents a fundamental principle for assessing the reliability of power systems in static security analysis. Existing studies mainly rely on centralized single-agent reinforcement learning frameworks, where centralized control is difficult to cope with regional autonomy and communication delays. In high-dimensional state–action spaces, these approaches often suffer from low efficiency and unstable policies, limiting their applicability to large-scale grids. To address these issues, this paper proposes a Multi-Agent Deep Reinforcement Learning (MADRL) method enhanced with Curriculum Learning (CL) and Prioritized Experience Replay (PER). The proposed framework adopts a Centralized Training with Decentralized Execution (CTDE) paradigm, where independent agents are assigned to different system regions to enable autonomous decision-making and interregional coordination. In addition, the Actor–Critic (AC) architecture is refined with optimized value update rules to mitigate Q-value overestimation. A curriculum learning mechanism based on source–load fluctuation intensity further guides agents from simple to complex operating conditions, enhancing convergence and policy robustness. Simulation results on the IEEE 39-bus system demonstrate that the proposed method efficiently generates coordinated multi-region control strategies, eliminates voltage and current violations under N-1 contingencies, and consistently outperforms the baseline MADRL approach in terms of decision performance and robustness under fluctuating source–load scenarios.

KEYWORDS: Multi-agent deep reinforcement learning; static security analysis; preventive control; curriculum learning; N-1 guidelines

1 Introduction

Static security analysis (SSA) is a fundamental approach to ensuring the secure and stable operation of power systems. Over time, SSA has evolved into a mature theoretical framework with established engineering standards. As the core criterion for static security assessment, the N-1 criterion can provide decision-making support in system accident prevention and operational stability improvement [1]. When the system operating point violates the N-1 security boundary, preventive or corrective measures must be implemented to maintain system security [2].

At present, static security prevention and control primarily rely on model-driven approaches, where mathematical models and optimization algorithms are employed to derive the optimal control strategy under safety constraints [3–8]. However, this type of method is highly dependent on the precise modeling of the alternating current power flow equations and the complete acquisition of line parameters. With the expansion of the scale of the power system, this method encounters significant challenges in terms

of computational efficiency and real-time applicability. To address these issues, data-driven approaches have emerged, offering nonlinear modeling capabilities, reduced dependence on physical models for SSA. However, Deep Reinforcement Learning (DRL) has the advantages of eliminating the need for precise physical models and supports end-to-end decision-making, and has shown good application prospects in power system optimization scheduling [9–11], emergency control [12–14] and other fields. Studies have shown that DRL can drive agents to learn control strategies through reward function design, enabling generator output adjustment, load shedding, and other corrective operations to eliminate security risks. For example, [15] achieves the accurate calculation of the power adjustment amount based on Deep Q Network(DQN) and sensitivity analysis; the literature [16] constructs a trend adjustment Markov decision-making process that satisfies the static stability constraints, and achieves the collaborative control of generator action and multi-objective under the framework of parallel DRL; the literature [17] quickly generates an approximate optimal adjustment strategy considering transient stability constraints through distributed Deep Deterministic Policy Gradient (DDPG); the literature [18] uses the PQ decoupling characteristics to propose a dual-agent collaborative architecture to realize the coordinated control of active power and voltage.

However, most of these methods are based on a centralized single-agent framework, which face two major challenges in large-scale systems: (1) centralized control makes it difficult to guarantee communication latency or ensure regional autonomy; and (2) the high-dimensional state–action space increases training complexity, often preventing effective convergence. In this context, Multi-Agent Deep Reinforcement Learning (MADRL) has become a promising direction due to its ability of distributed decision-making and global coordination. By decomposing centralized control into a cooperative game among agents, each agent can make independent decisions under a decentralized structure and achieve global coordination through local interaction. This approach not only relieves computational pressure caused by high dimensionality but also better fits the autonomy and coordination needs of large-scale grids. Studies have verified the advantages of MADRL in power system control. For example, Reference [19] proposes a two—stage control method combining optimized scheduling and Multi-Agent Deep Deterministic Policy Gradient (MADDPG) for vol/var optimization; Reference [20] formulates multi-regional volt/var coordination as a partially observable Markov game and proposes a robust regional collaborative VVC strategy; Reference [21] develops a centralized training–decentralized execution multi-agent voltage control framework under N-1 conditions; Reference [22] employs LSTM-enhanced MADDPG for microgrid frequency regulation under renewable uncertainty.

Although MADRL has demonstrated strong potential in improving the scalability and adaptability of power system control, the baseline MADDPG framework still faces challenges such as Q-value over-estimation, unstable convergence, and low sampling efficiency in high-dimensional environments [23]. Compared with other multi-agent reinforcement learning algorithms that rely on value factorization or shared advantage estimation, MADDPG offers a deterministic actor–critic structure capable of directly handling continuous control actions. This property is essential for preventive control in large-scale power systems, where generator voltage and power adjustments must satisfy nonlinear physical constraints and maintain high control precision. Discretizing these continuous actions, as required by many value-based algorithms, could reduce control granularity and impair the enforcement of N-1 static security limits.

Building on the advantages of MADDPG, this study focuses on enhancing its training stability and efficiency in complex multi-agent environments. A Curriculum Learning (CL) mechanism [24,25] is introduced to guide agents from simple to more challenging N-1 operating conditions, improving adaptability to varying source–load fluctuations. Additionally, a Prioritized Experience Replay (PER) mechanism [26] is integrated to emphasize high-value samples during centralized training, thereby accelerating convergence

and improving sample utilization. The resulting CL-PER-MADRL framework maintains the physical interpretability of control actions while achieving more stable and efficient policy learning for preventive control tasks in large-scale power systems.

Based on this, this research proposes a multi-agent actor-critic learning framework integrating CL and PER for static security prevention and control of power systems. This approach refines the value update rules to alleviate Q-value overestimation and incorporates progressive learning to guide agents from simple to complex scenarios, thereby improving convergence and policy robustness.

The main contributions of this article are as follows:

1. A curriculum-learning-guided prioritized replay mechanism is designed to improve convergence efficiency and robustness to source–load fluctuations without requiring precise physical modeling.
2. A distributed control framework based on CTDE in static security prevention and control is developed, effectively solve the problems of dimensional disasters, communication delays and single points of failure, and realize regional autonomy and cross-regional collaboration.
3. On the basis of the traditional AC architecture, in view of the problem of overestimation of Q value in multi-agent training, an optimized value update rule is proposed, which enhances the stability and control effect of the strategy.

2 Static Security Prevention and Control Modeling of the Power System

In power system studies, the primary criterion for static security analysis is that, following any N-1 contingency (i.e., the disconnection of a single component), system equipment should not become overloaded and bus voltages must remain within permissible limits. Preventive and corrective control are generally defined as adjustments to the system's operating point to ensure adequate safety and stability margins during normal operation. The basic process of static security prevention and control typically involves modifying the operating point of the system, sequentially disconnecting transmission lines, calculating bus voltages and branch active power flows, and checking whether any violations occur under N-1 contingency conditions. Based on these principles, the static security prevention and control optimization model can be formulated as follows:

(1) The control variables are the generator's active power and terminal voltage.
(2) Constraints include both equality and inequality constraints. The equality constraints correspond to the power flow equations after the control variables are applied, as shown in Eq. (1)

$$\begin{cases} P_i = V_i \sum\limits_{j \in N} V_j \left(G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij} \right) \\ Q_i = V_i \sum\limits_{j \in N} V_j \left(G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij} \right) \end{cases} \tag{1}$$

where P_i and Q_i denote the active and reactive power injected at bus i, respectively; V_i represents the voltage magnitude at bus i; θ_{ij}, G_{ij} and B_{ij} are the voltage phase angle difference, conductance and admittance between bus i and j, respectively.

The inequality constraints specify the upper and lower bounds of the control variables during the adjustment process, expressed as:

$$P_{G,i,\min} \le P_{G,i} \le P_{G,i,\max} \tag{2}$$

$$\Delta P_{G,i,\min} \le \Delta P_{G,i} \le \Delta P_{G,i,\max} \tag{3}$$

$$Q_{G,i,\min} \le Q_{G,i} \le Q_{G,i,\max} \tag{4}$$

where $P_{G,i}$, $P_{G,i,\max}$ and $P_{G,i,\min}$ denote the active power of the synchronous generator i and its upper and lower limits; $\Delta P_{G,i}$, $\Delta P_{G,i,\max}$ and $\Delta P_{G,i,\min}$ denote the active power variation of synchronous generator i and its maximum upward and downward ramping capability; $Q_{G,i}$, $Q_{G,i,\max}$ and $Q_{G,i,\min}$ are the reactive power of synchronous generator i and its upper and lower limits.

(3) The control objective is to minimize both the magnitude and number of voltage, current, and power violations in the static security analysis, as represented in Eq. (5).

$$\min\left(J_V + J_I + J_S\right) \tag{5}$$

Specifically:

$$J_V = \sum_{k=1}^{\Omega_L} \sum_{i\in\Omega_B} \left[\max\left(V_i^k - V_i^{\max}, 0\right)^2 + \max\left(V_i^{\min} - V_i^k, 0\right)^2\right] \tag{6}$$

$$J_I = \sum_{k=1}^{\Omega_L} \sum_{(i,j)\in\Omega_L} \max\left(|I_{ij}^k| - I_{ij}^{\max}, 0\right)^2 \tag{7}$$

$$J_S = \sum_{k=1}^{\Omega_L} \sum_{m\in\Omega_S} \max\left(|S_m^k| - S_m^{\max}, 0\right)^2 \tag{8}$$

where Ω_L, Ω_B and Ω_S denote the sets of branch, bus and equipment apparent power, respectively; V_i^k, I_{ij}^k, and S_m^k denote the bus voltage magnitude, line current magnitude, and equipment apparent power, respectively, under the k-th line outage; $V_i^{\max}$, $V_i^{\min}$, $I_{ij}^{\max}$, and $S_m^{\max}$ are their corresponding limits.

Each term in the objective function represents the cumulative violation magnitude of a specific operational variable under all N-1 contingencies. Since all violations are expressed as squared exceedance values normalized by their respective operational limits (e.g., $V_i^{\max}$, $I_{ij}^{\max}$, $S_m^{\max}$), the overall objective function becomes dimensionless. Equal weighting is applied among the three components J_V, J_I, and J_S, ensuring balanced consideration of voltage, current, and apparent power violations without introducing subjective prioritization. This formulation highlights comprehensive static security improvement rather than emphasizing a single type of violation.

In summary, static security preventive control in power systems primarily relies on regulating generator output. A nonlinear optimization model is constructed with power flow equations as equality constraints and equipment operation limits as inequality constraints. However, as the system scale expands and the number of N-1 contingency increases sharply, the model becomes highly nonlinear and computationally intensive, making traditional optimization-based methods difficult to apply in real time.

To address these challenges, this study introduces a MADRL framework to enhance the adaptability and real-time performance of static security preventive control.

3 Static Security Prevention and Control Method Based on MADRL

In order to enhance the real-time performance of static security preventive control in power systems, this study employs the MADRL approach to accelerate decision-making. Specifically, the power grid is partitioned into regions, with each region assigned an agent responsible for regulating generator outputs; based on the multi-agent Actor–Critic framework, an improved training mechanism is developed to enable the agents to jointly learn regulation schemes that satisfy the N-1 security constraints of the entire system.

It should be noted that the partitioning strategy of the system directly affects the distribution characteristics of adjustable units in each region, which in turn influences the convergence and stability of multi-agent

training. The algorithm examined in this study assumes that the partitioning is fixed according to the network topology and permits the sharing of state and action information among agents during the centralized training phase to ensure effective learning and coordination of the scheduling scheme.

3.1 Distributed Partially Observable Markov Decision Processes

Before applying the MADRL algorithm, the problem is formulated as a distributed partially observable Markov decision process (Dec-POMDP). In this framework, the static security preventive control of the power system is represented as a six-tuple: $(S, O_i, A_i, R_i, P, \gamma)$, where S is the system state space; O_i is the local observation set of agent i, containing only partial information from the global state; A_i is the action set of agent i; R_i is the reward function of agent i, which satisfies $S \times A \rightarrow R$; P is the state transfer function; and γ is the discount factor. At each time step t, agent i observes the operational state o_i^t of its region and selects an action according to its policy $\pi_i\left(a_i|o_i\right)$. The system then transitions to the next state $s^{t+1} \sim P\left(s^{t+1}|s^t, a^t\right)$ based on the current state s^t and the joint actions a^t of all agents, providing immediate rewards r_i^t to each agent. Subsequently, each agent receives a new local observation and stores the current state, action, reward, and next state in the replay buffer for network updates. This process repeats until a termination condition is met and a training episode is completed.

(1) The observable state of the agent

Due to limitations such as communication constraints and privacy protection, each agent's observation is restricted to its local measurement data. Each agent makes decisions solely based on its local state, without requiring complex communication devices to obtain information from other agents. The observable state is defined in Eq. (9).

$$o_i = \{\eta_l, P_G, V_n, P_{load}, Q_{load}\} \tag{9}$$

(2) The action of the agent

The range of real-time control decision variables is mapped to the interval [−1, 1] according to the generator ramping constraint in Eq. (3), and the mapped variables constitute the agent's action space.

$$a_i = \{\Delta P_G, \Delta U_G\} \tag{10}$$

where ΔP_G And ΔU_G denote the generator active power and the change in terminal voltage, respectively.

(3) Rewards for agents

The control objective of each agent is to minimize violations of static security constraints. The reward generation process is illustrated in Fig. 1. After the joint actions of all agents are executed in the environment, system security constraints under normal conditions are first evaluated, followed by N-1 contingency verification. The resulting immediate reward is then fed back to each agent. When the trend does not converge, a greater negative reward K_1 is given.

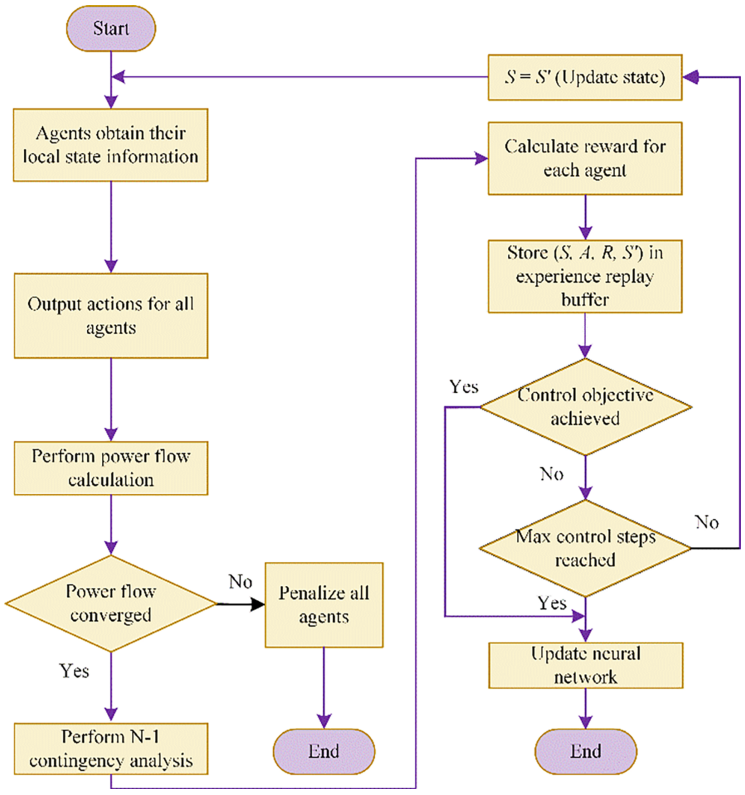


Figure 1: Reward generation process

The training goal of MADRL is to find the optimal strategy to maximize the cumulative return. To this end, the static security overstepping problem is transformed into a penalty item and a reward function is introduced. The reward function design is shown in Eq. (11):

$$r_i = \begin{cases} -K_1 & \text{Post-decision power flow non-convergence} \\ -K_2 & \text{Power flow fails to converge for an N-1 contingency} \\ F_{add} - F_I - F_{U_k} - K_3 - K_4 & \text{Limit violations occur during the N-1 contingency analysis} \\ 1.5 & \text{The system passes all N-1 contingency checks without any violations} \end{cases} \tag{11}$$

where K_1 is the penalty for current non-convergence after agent i's action; K_2 is the penalty for non-convergence during the N-1 process; K_3 is the penalty for violations of balancing machines; K_4 is the penalty for violations of regulating units; F_{add} is the positive reward assigned when the total number of voltage over-limit $F_v(k)$ and current over-limit $F_i(k)$ in the system is small after the static security verification, as shown in Eq. (12); F_I and F_{U_k} are the penalties for the agent's violation of static security constraints.

$$F_{add} = \frac{K_5}{F_v(k) + F_i(k)} \tag{12}$$

where K_5 denotes the reward coefficient.

The reward function is designed to address the requirements of static security preventive control in power systems, combining a severity measurement method based on utility theory with a discrete reward

mechanism to guide agents in accurately perceiving and responding to static security risks. The penalty items F_{IU_k} and F_{U_k} for each agent's violation of static security constraints are defined as follows:

$$F_I = \sum_{k=1}^{\Omega_L} \{\min[a(e^{f_{I_k}} - 1), K_6], K_6]\} \tag{13}$$

$$F_{U_k} = \sum_{k=1}^{\Omega_L} \{\min[a(e^{f_{U_i}} - 1), K_6], K_6]\} \tag{14}$$

Among them:

$$f_{U_i} = \begin{cases} U_{\min} - U_i, U_i < U_{\min} \\ U_i - U_{\max}, U_i > U_{\max} \\ 0, \text{ other} \end{cases} \tag{15}$$

$$f_{I_k} = \begin{cases} I_k - I_{k,\max}, I_k > I_{k,\max} \\ 0, \text{ other} \end{cases} \tag{16}$$

In summary, the reward function consists of three distinct components that guide the learning process. The penalty terms, governed by coefficients $K_1 - K_4$, serve to penalize power flow non-convergence and operational constraint violations, ensuring all actions yield feasible system states. The guidance reward F_{add} provides dense, positive feedback for control actions that reduce the number of security violations. Furthermore, a significant sparse reward is granted only when a fully secure state (zero violations) is achieved, collectively steering the agent toward robust N-1 static security compliance.

3.2 MADDPG Algorithm

In order to solve the problem of instability in multi-agent environments, Multi-Agent Deep Deterministic Policy Gradient(MADDPG) adopts a centralized training-decentralized execution framework; in the centralized training phase, each regional agent first observes the regional state information, feeds it into its policy network to generate actions, and then all actions are combined to act on the environment. The environment feeds back the rewards of each agent and transfers them to the next operating state, and at the same time stores the experience in the experience replay buffer. In the update phase, small batches of data are randomly sampled from the experience replay buffer and aggregated into joint experiences (s, a, r, s'). When the Critic network is updated, it will consider the other agents' policies and actions, and use the joint experience to update the network parameters, so as to achieve coordinated decision-making between agents. The Actor network updates its policy based on the Q-value given by the Critic network. After the network update is completed, each agent in the online execution phase makes decisions solely based on local observations, and there is no need to communicate with other agents. This method takes into account the communication restrictions in the actual scenario, making it suitable for the coordinated prevention and control of complex power systems in the sub-regions.

If the environment contains N agents, the system maintains 2N Actor–Critic networks. Among them, the policy of the i-th agent is denoted as π_i, with parameters θ_i. The cumulative expected return of the i-th agent is given by:

$$J(\theta) = E[R_i] = E_{s-p^{\pi}, a_i - \pi_{\theta i}} \left[\sum_{i=0}^{\infty} \gamma^t r_i, t \right] \tag{17}$$

Then the deterministic strategy gradient of the i-th agent is expressed as:

$$\nabla_{\theta_i} J(\mu_i) = E_{o,a\sim D}\left[\nabla_{\theta_i}\mu_i(a_i|o_i)\,\nabla_{a_i} Q_i^{\mu}(s, a_1, \ldots, a_N)\big|_{a_i=\mu_{\theta_i}(o_i)}\right] \tag{18}$$

where o_i denotes the local observation of the i-th agent; D is the experience replay buffer, which stores the experience trajectories of all agents, and each sample is a quadruple, where: $s = [o_1, o_2, \ldots, o_n]$, $a = [a_1, \ldots, a_N]$, $r = [r_1, r_2, \ldots, r_n]$; Q_i^{μ} is the joint action value function, that is, a centralized state–action value function, each sample not only contains the local observation state and the execution action, but also incorporates the state-action information of other agents. This data structure effectively addresses the technical bottlenecks of traditional reinforcement learning in the field of multi-agents, so that agents can not only perceive changes in their own state, but also obtain global action strategies. Through the state-action joint coding in the process of continuous strategy update, the Markov nature of environmental interaction is preserved, and the problem of environmental instability caused by partial observability of traditional methods is effectively mitigated. Even if the strategy is continuously updated, the environment maintains stability. The update of Q_i^{μ} is updated according to Eq. (19):

$$L(\theta_i) = E_{s,a,r,s'}\left[Q_i^{\mu}(s, a_1, \cdots, a_N) - y_i)^2\right] \tag{19}$$

where y_i is obtained by Eq. (20):

$$y_i = r_i + \gamma \cdot Q_i^{\mu'}(s', a_1', \ldots, a_N')\big|_{a_i'=\mu_i'(o_i')} \tag{20}$$

3.3 Improved Actor-Critic Strategy

In the MADDPG algorithm, the target actor network and the target critic network adopt a "soft update" strategy. This strategy can make the current network too similar to the target network, which hampers the effective separation of action selection from policy evaluation. It can also lead to an excessive pursuit of maximizing long-term discount returns, resulting in overestimation of the true Q-value. With multiple rounds of iterative updates, this overestimation tends to accumulate during policy exploration, increasing both deviation and variance. Consequently, the agent's ability to explore globally optimal solutions and make optimal decisions in the current state may be impaired, which may eventually undermine the effectiveness of the learning strategy.

In order to solve overestimation in DRL, this paper incorporates the AC framework from the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm into MADDPG. Two independent critic networks Q_1^{μ} and Q_2^{μ} are introduced, along with their corresponding target networks $Q_1^{\mu'}$ and $Q_2^{\mu'}$. The minimum of the two Q-value estimates is used as the target value. This approach effectively reduces overestimation deviation, thereby improving online learning stability. The Q-value calculation in Eq. (20) is therefore replaced by the formulation in Eq. (21).

$$y_i = r_i + \gamma \cdot \min_{m=1,2} Q_{i,m}^{\mu'}(s', a_1', \ldots, a_N')\big|_{a_i'=\mu_i'(o_i')} \tag{21}$$

3.4 Importance Sampling

Prioritized Experience Replay (PER) speeds up convergence by prioritizing experiences with large temporal-difference (TD) errors, which are more informative for learning. However, such prioritization introduces a sampling bias, which is corrected by applying importance sampling weights Eqs. (25) and (26) to ensure unbiased gradient estimation.

In the MADDPG framework, the update of the action-value function $Q(s,a)$ depends on the gradient transfer of the TD error. The TD error directly reflects the discrepancy between the agent's estimated and actual experience values. Based on this characteristic, this study uses $|\delta|$ as a quantitative indicator of the importance of experience, as shown in Eq. (22):

$$\delta = \left(y - Q^{\mu'}\left(s, a_1, a_2, \cdots, a_N\right)\right)^2 \tag{22}$$

The larger the δ, the greater the discrepancy between the target network's prediction and the actual experiential return. The sampling frequency of such experiences should be increased to accelerate convergence between the target network and the current network, thereby improving training efficiency. Therefore, the sampling probability of each experience is defined as Eq. (23), with its priority given by Eq. (24).

$$P(j) = \frac{p_j^a}{\sum_k p_k^a} \tag{23}$$

$$p_j = |\delta_j| + \varepsilon \tag{24}$$

In Eq. (23): When $\alpha = 0$, it corresponds to uniform sampling. In Eq. (24), $\varepsilon \in (0,1)$ is a small constant that avoids probability collapse. Meanwhile, the TD error of newly added experiences is initialized as $\delta_{\max}$.

Although PER improves learning efficiency by replaying experiences with high TD errors more frequently, it can also lead to data distribution deviations and training instability. Oversampling of high-priority samples changes the empirical data distribution, resulting in a biased gradient expectation during policy evaluation. To address this issue, importance sampling is introduced to compensate for the non-uniform sampling probabilities while maintaining the convergence properties of stochastic gradient descent. The corresponding importance sampling weight formula is shown in Eq. (25).

$$\omega_j = \left(\frac{1}{N} \cdot \frac{1}{P(j)}\right)^{\beta} \tag{25}$$

where N is the current capacity of the replay buffer, and $P(j)$ is the sampling probability, and β is the weight coefficient, which controls the influence of the importance sampling weight in the learning process, and linearly anneals to 1 over the course of training. In order to improve the stability during convergence, the weights are normalized to stabilize the gradient update scale and suppress oscillations during training. Therefore, the Eq. (25) is normalized to obtain the formula:

$$\omega_j = \frac{(N \cdot P(j))^{-\beta}}{\max_i(\omega_i)} \tag{26}$$

In the MADDPG algorithm, the parameters of the strategy network depend on the selection of the value network, and the parameters of the value network are updated by the loss function of the value network. Combined with the above-mentioned importance experience playback and improved Actor-Critic strategy, the loss function in Eq. (19) is modified to Eq. (27):

$$L_{PER}(\theta_i) = \frac{1}{2}\sum_{m=1}^{2} E_{s,a,r,x'}\left[\omega_j \cdot \left(Q_{i,m}^{\mu}\left(s, a_1, \cdots, a_N\right) - y_i\right)^2\right] \tag{27}$$

This section thus incorporates prioritized experience replay with importance sampling into the proposed CTDE-based multi-agent framework. The implementation process and network structure of the algorithm proposed by this research institute are shown in Fig. 2.

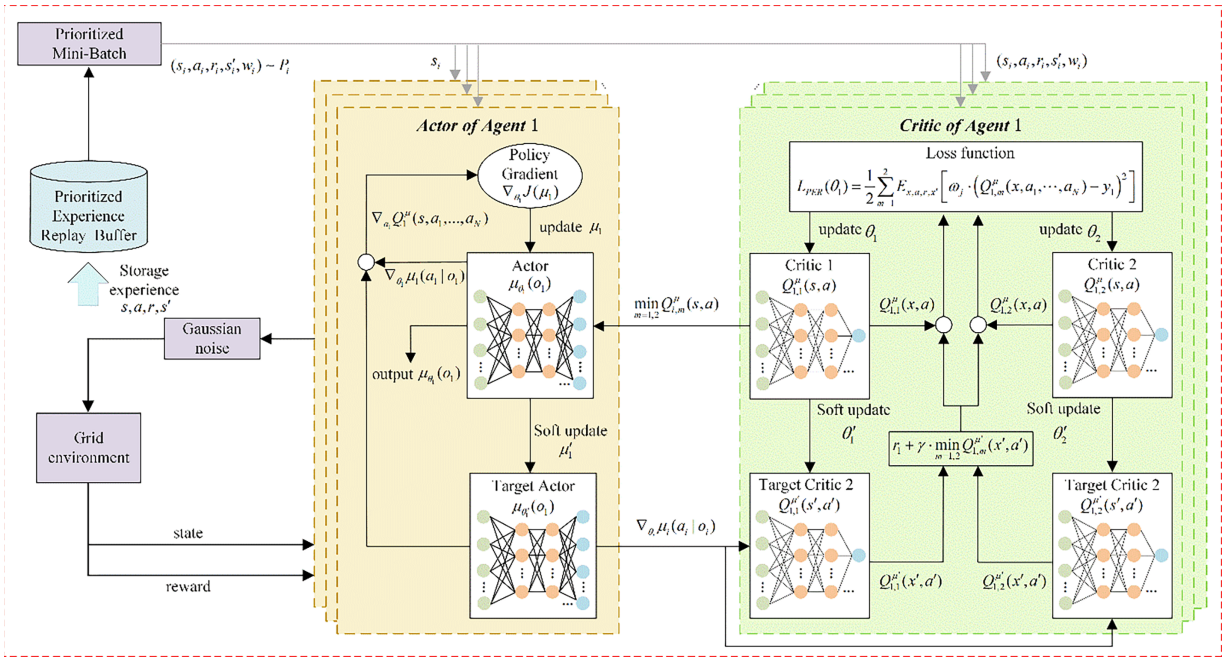


Figure 2: The implementation process and network structure configuration of the algorithm proposed in this study

3.5 Curriculum Learning

In the reinforcement learning framework, Curriculum Learning (CL) has emerged as an effective paradigm to improve the efficiency of policy learning. Its core principle lies in organizing training tasks in a "from simple to complex" manner, thereby guiding agents to progressively acquire stable and efficient decision-making capabilities. In each subsequent stage, load and generator disturbances are gradually increased, enabling the agents to learn from basic stability control to complex system coordination. By structuring the learning environment with progressive levels of difficulty, agents can rapidly acquire fundamental control strategies under low-complexity conditions and subsequently adapt to increasingly challenging scenarios. Through this gradual exposure, the learned policies not only enhance adaptability to multi-source load fluctuations while maintaining robustness and stability under severe disturbances.

In this work, CL is integrated into the preventive control training process of power systems. At the early stage, the focus is placed on ensuring basic stability control, allowing agents to understand and internalize the fundamental operational dynamics of the grid. During the intermediate stages, progressively broader load variations and disturbance conditions are introduced, thereby enhancing policy transferability and generalization. Ultimately, the agents are trained to respond rapidly and reliably to large-scale disturbances within highly complex operating environments, thereby achieving both robustness and practical applicability. This hierarchical design ensures stability and efficiency during training and establishes a solid foundation for subsequent optimization of cooperative control strategies.

To provide an intuitive overview of the proposed training framework, Algorithm 1 summarizes the pseudocode of the CL-PER-MADRL algorithm. This algorithm integrates CL with PER within an MADRL

framework. The pseudocode illustrates the key training process, detailing the curriculum-based progression with increasing difficulty levels and the specific performance criteria required to advance to the next stage.

Algorithm 1: CL-PER-MADRL for N agents

Divide training tasks into n stages with increasing difficulty
Initialize replay buffer D, exploration noise ε, curriculum level $j \leftarrow 1$
for episode = 1 **to** M **do**
  Obtain initial system state s
  for $t = 1$ **to** T **do**
    for each agent $i = 1$ **to** N **do**
      Select action $a_i \leftarrow \pi_i(s_i) + \varepsilon_i$
    end for
    Execute joint action $a \leftarrow (a_1, \ldots, a_N)$
    Environment returns reward r_i and next state s'
    Store transition (s, a, r, s') into replay buffer D
    $s \leftarrow s'$
  end for
  for each agent $i = 1$ **to** N **do**
    Sample a mini-batch of size S from D according to Eqs. (24) and (26)
    Update Critic network parameters according to Eq. (27)
    Update Actor network parameters according to Eq. (18)
  end for
  for each agent $i = 1$ **to** N **do**
    Soft update target networks: $\theta_i \leftarrow \tau\theta_i + (1 - \tau)\theta_i'$
  end for
  if average control success rate over last 50 episodes > 95% **then**
    $j \leftarrow j + 1$
  end if
  if $j = n$ and average control success rate over last 50 episodes > 95% **then**
    Break
  end if
end for

4 Example Analysis

4.1 Experimental Scene Setup

In this paper, the proposed method is simulated and verified by the IEEE 39-bus transmission system which is partitioned into three regions, as shown in Fig. 3. The observation space of each agent includes the active power of the generator, the line load rate, the bus voltage magnitude, and active and reactive power at load buses. The action space is defined as a continuous high-dimensional space for the adjustment of the generator's output, and its adjustment range is constrained to 60%–100% of the rated output. During training, seven load levels (85%, 90%, 95%, 100%, 105%, 110%, and 115% of the reference load) are considered, with random source–load disturbances applied. Static N-1 verification requires traversing all transmission lines.

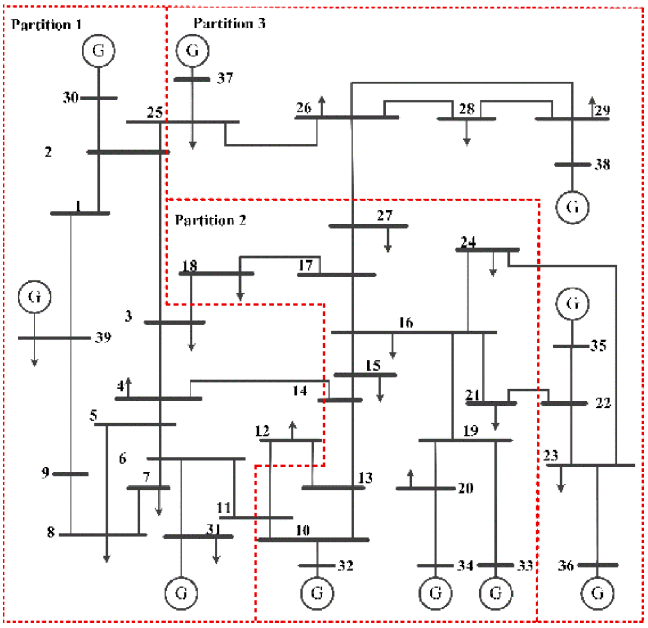


Figure 3: Schematic diagram of IEEE 39-bus partition

4.2 Agent Network Architecture and Parameter Settings

Based on the results of power grid partitioning, a distributed decision-making architecture is developed, where an independent multi-layer fully connected neural network is assigned to each region as the agent model, enabling autonomous regulation and decision-making. The number of neurons in the input layer of the agent actor network corresponding to regions 1, 2, and 3 corresponds to the dimensionality of the regional operating state. The output is the active power of the generator in each region and the adjustment of terminal voltages with the tanh function used as the activation function. Based on experimental evaluation, the number of hidden layers in both the actor and critic networks is set to three; the fully connected layers use the ReLU activation function, and the number of neurons in each layer is (512, 512, 256) for the actor and (1024, 512, 256) for the critic. Each neural network is updated once after every load adjustment. Both networks adopt the Adam optimizer for training, and Gaussian noise is employed for exploration. The specific parameter settings are summarized in Table 1.

Table 2 presents the components that exceed operational limits in the static N-1 verification conducted prior to adjustment. Static N-1 verification requires traversing all transmission lines individually; for each contingency, the system is checked for limit violations. The columns report the total number of violations accumulated over all N-1 contingencies: Voltage violations, Current violations, and Slack generator violations respectively denote the number of buses, transmission lines, and slack generator power outputs that exceed permissible limits in any contingency. Numerous voltage, current, and slack generator elements in the system exceed permissible thresholds, thereby violating static security requirements. Therefore, it is necessary to adjust the system operating point by applying preventive control and related measures to ensure static security compliance.

Table 1: Parameter settings

Parameters	Value
Batch size	128
Learning rates of the actor network	0.0001
Learning rates of the critic network	0.0003
τ	0.01
γ	0.97
K_1	2
K_2	0.04
K_3	0.004
K_4	0.01
K_5	0.000004
K_6	0.0006

Table 2: Component overruns after static security verification of the system before adjustment

Load level	Voltage violations	Current violations	Slack generator violations	Total
0.85	94	253	36	383
0.9	32	217	36	285
0.95	5	188	36	229
1.0	1	152	36	189
1.05	1	95	34	130
1.1	1	40	34	75
1.15	0	12	0	12
Total	134	957	212	1303

4.3 The Results of Model Training

The power flow calculation program of the simulation example system, as illustrated in Fig. 3, is developed on the Pandapower platform, which provides the simulation environment for multi-agent training. The proposed algorithm is implemented using PaddlePaddle and is employed to train the multi-agent model. The hardware platform consists of an NVIDIA RTX A4000 GPU, an Intel(R) Core(TM) i7-13700 CPU, and 16 GB of RAM.

During the training process of the MADRL agent, each load is explored 12 times per round, and the power system is statically checked for safety after each current adjustment. The training process adopts intensive CL to reduce the difficulty of training, and the course is divided into four stages for learning. The training scenario settings for each stage are shown in Table 3.

The regional agents in this study are fully cooperative, and the total cumulative rewards obtained by the three agents in each round are shown in Fig. 4. Both the reward curve and the actor–critic loss converge smoothly, indicating that the policy has reached a stable equilibrium. The total training duration for the proposed MADRL model was approximately 1 h and 55 min. To ensure reproducibility, random seeds were fixed for all experiments, including neural network initialization, noise generation, and environment randomness.

Table 3: Scenario setting for each stage of training

Training stage	Load variation interval (%)	Load variation interval (%)	Generator disturbance (%)
Phase I	[0.85, 1.15]	1	5
Phase II	[0.85, 1.15]	2.5	5
Phase III	[0.85, 1.15]	2.5	10
Phase IV	[0.85, 1.15]	2.5	15

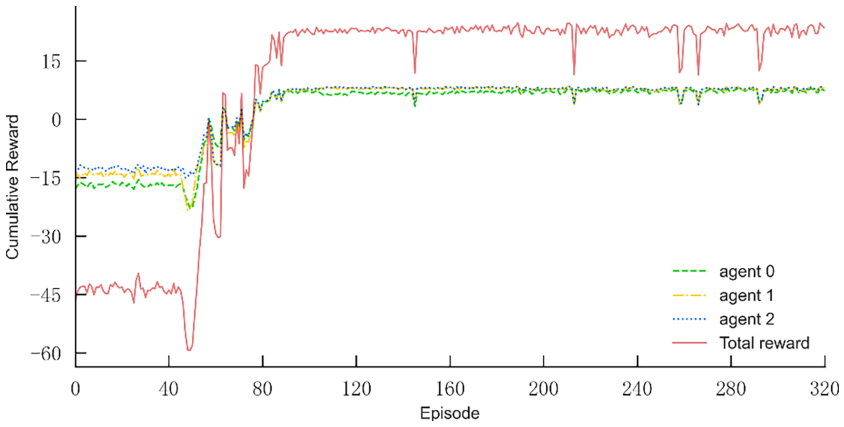


Figure 4: Cumulative rewards of three regional agents and their total reward

In the first stage, the weight parameters of each neural network are randomly initialized, which fluctuate greatly during early training but gradually converge as training progresses. During the first 45 rounds, only the experience buffer was filled without updating the network, resulting in low reward values. From rounds 45 to 100, network updates commenced, causing the reward curve to rise rapidly as the agent gradually learned to satisfy the static security and stability requirements of preventive control measures. At round 140, the training entered the second stage. During the second stage, the load fluctuation range was increased, yet the control success rate remained high, indicating that the proposed MADRL method exhibits strong generalization ability. At round 190, the third stage commenced. The substantial increase in initial generator output disturbance caused the agent to partially exceed limits at the beginning of this stage; however, it stabilized after training. Upon entering the fourth stage, the agent continued to maintain high control performance even under greater disturbance intensities.

To further verify the robustness of the learning framework to hyperparameter variations, an additional experiment was performed in which the learning rate was increased by 20% compared to the value listed in Table 1. The resulting reward curve, as shown in Fig. 5, maintained a smooth and stable convergence pattern similar to the original setting, indicating that the proposed MADRL algorithm exhibits strong robustness and stability against moderate parameter fluctuations.

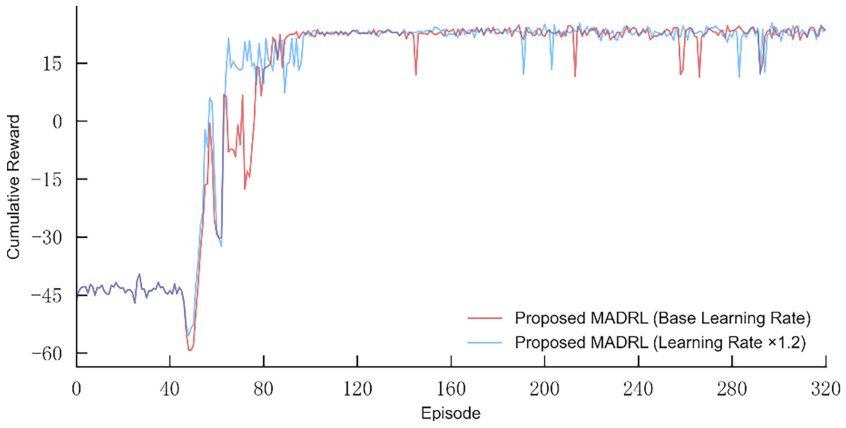


Figure 5: Reward convergence curves of the proposed MADRL under different learning rates

4.4 Model Validation and Performance Evaluation

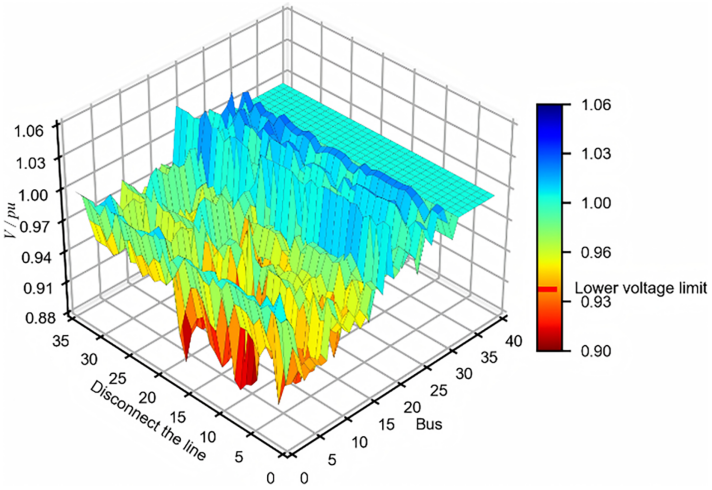
Before validating the proposed MADRL-based preventive control strategy, a conventional Optimal Power Flow (OPF) method was tested under the same system configuration to examine whether traditional optimization can ensure static security compliance. Although the OPF successfully converged to an economically optimal operating point, the resulting solution failed to satisfy N-1 static security criteria. Across multiple load levels, the OPF results exhibited dozens of cumulative voltage and current limit violations after N-1 contingency verification, indicating that OPF-based optimization, which does not explicitly consider contingency constraints, cannot guarantee static security in large-scale grid operations. This highlights the necessity of developing intelligent preventive control approaches that explicitly account for N-1 contingencies.

The effectiveness of the proposed algorithm in enhancing the static security of the power system was evaluated by deploying the trained agents in a simulation environment. During the testing phase, each agent performs real-time control actions based on the grid's operating state under the specified load conditions.

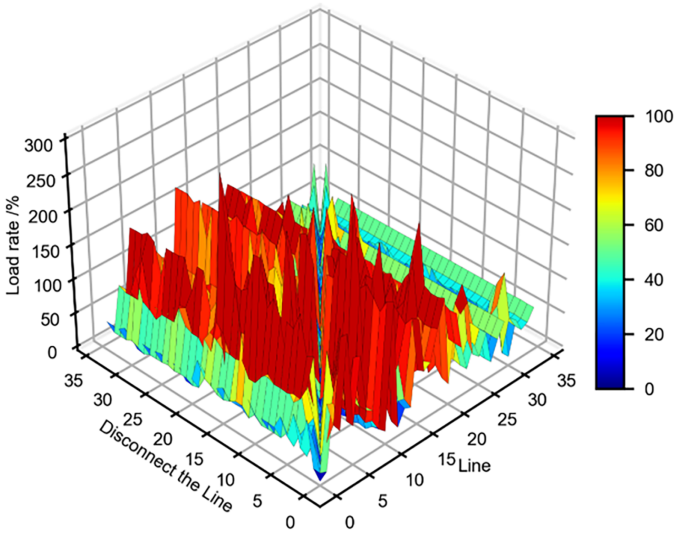
As an illustrative case, at a load level of 85%—which represents the most severe limit violations—Figs. 6 and 7 illustrate the bus voltages and line loading rates before and after control through N-1 contingency verification. The results indicate that initially, multiple bus voltages and line currents exceeded their operational limits, met static security requirements. Specifically, the lower voltage limit was violated 94 times, and current overloads occurred 253 times. When line 14–15 were disconnected, the voltage at bus 14 dropped to 0.874 p.u., while the load rate of line 2–3 peaked at 308%. clearly indicates static insecurity.

Specifically, at this load level, the overlimit conditions summarized in Table 4 required five consecutive control adjustments to restore static security, with N-1 verification conducted after each adjustment of the system operating point. Through progressive tuning of operating points, the system gradually approached compliance with static security standards. Specifically, after the first adjustment, the removal of line 14–15 reduced the peak load rate of lines 2–3 from 308% to 258% (a 16.2% decrease); the minimum voltage at bus 14 increased from 0.874 p.u. to 0.916 p.u. (a 4.81% increase), with no overvoltage observed. After the second adjustment, the load rate of line 2–3 further decreased to 214%, and the voltage at bus 14 increased to 0.974 p.u. marking the first time all bus voltages satisfied safety limits. After the third adjustment, lines 3–4 were removed, shifting the overload-dominant line to lines 5–10 (load rate of 147%), while all bus voltages remained within safe limits. Following the fourth adjustment, lines 14–15 and 2–3 were identified as key overloaded lines (132%), with voltage levels remaining within normal limits. After the fifth adjustment, N-1

verification confirmed the elimination of equipment overloads and voltage violations, indicating that static security had been fully restored.

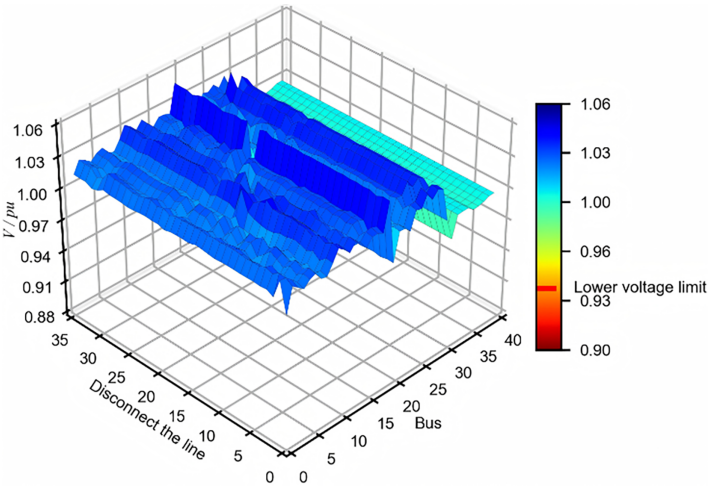


(a) Bus voltage under light-load conditions

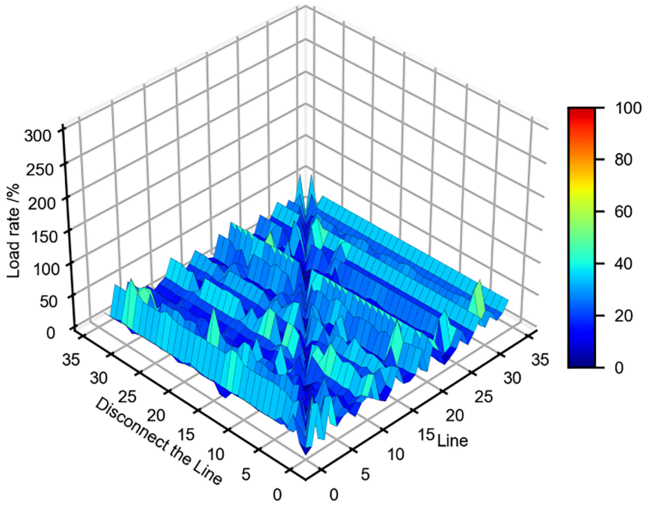


(b) Line load ratio under light-load conditions

Figure 6: Bus voltage and line load ratio before control



(a) Bus voltage under light-load conditions



(b) Line load ratio under light-load conditions

Figure 7: Bus voltage and line load ratio after control

Table 4: Overlimit situation during light load level regulation

Load level 0.85	Voltage violations	Current violations	Slack generator violations	Total
Before adjustment	94	253	36	0
1st adjustment	10	196	36	0
2nd adjustment	0	94	36	0
3rd adjustment	0	21	36	0
4th adjustment	0	3	34	0
5th adjustment	0	0	0	0

After completing training, the agent was deployed to evaluate each typical load condition. The results indicate that under various random operating scenarios, line outages—whether due to failures or maintenance—did not result in system voltages or currents exceeding operational limits.

4.5 Comparative Analysis of Algorithm Effectiveness

To verify the effectiveness of the proposed MADRL-based static security preventive control method, MADDPG was employed for comparison, using the fourth stage as the target training benchmark. Fig. 8 presents the total reward curves of agents for each algorithm. As shown in Fig. 8, the MADDPG reward curve exhibits substantial fluctuations, indicating poor performance across varying load and source fluctuation scenarios.

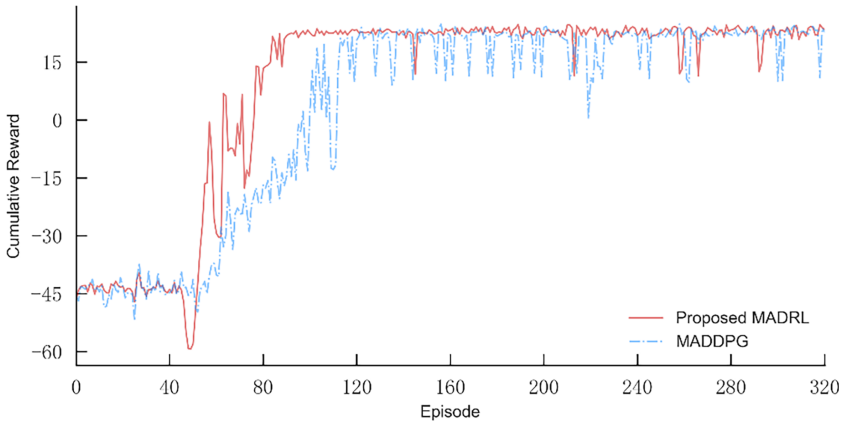


Figure 8: Comparison of the proposed MADRL against the baseline MADDPG on total cumulative rewards

The proposed method also began neural network updates at round 45 but incorporated a PER mechanism to prioritize under-learned experiences via importance sampling, thereby accelerating strategy convergence and enabling the agent to better extract local observation features and retain historical state information. Furthermore, the improved Actor-Critic network structure, combined with a well-designed curriculum, allows the agent to initially accumulate successful experiences in stable control under low disturbance conditions, facilitating a smooth transition as task difficulty gradually increases, and enabling the early-learned control strategies and features to transfer to more complex scenarios. This shallow-to-deep training approach effectively enhances strategy adaptability and robustness across varying load and source fluctuation conditions, maintaining high control performance under severe disturbances. Compared to direct training on challenging tasks, the proposed method substantially improves training stability and convergence speed, mitigates excessive reward fluctuations, and enhances strategy adaptability and robustness under varying disturbance intensities.

Fig. 9 depicts the cumulative number of N-1 limit violations for each algorithm following sequential load adjustments in each round. As illustrated, the proposed method significantly reduces limit violations after network updates, while maintaining a consistent downward trend as task complexity increases. These results demonstrate that the CL strategy effectively reduces static security constraint violations in complex disturbance environments and fosters a more robust multi-round control strategy.

To evaluate effectiveness, 1000 random scenarios from the fourth training stage were selected for testing and compared with the baseline MADDPG algorithm. Evaluation metrics comprised the success rate of static security preventive control, the average decision time per step, and the total number of agent decisions. Simulation results on the IEEE 39-bus system are summarized in Table 5. Results indicate that the proposed method achieves a preventive control success rate of 99.2%, significantly surpassing the 96.2% achieved by MADDPG. Moreover, the proposed reinforcement learning framework exhibits superior performance in both control efficiency and cross-scenario generalization.

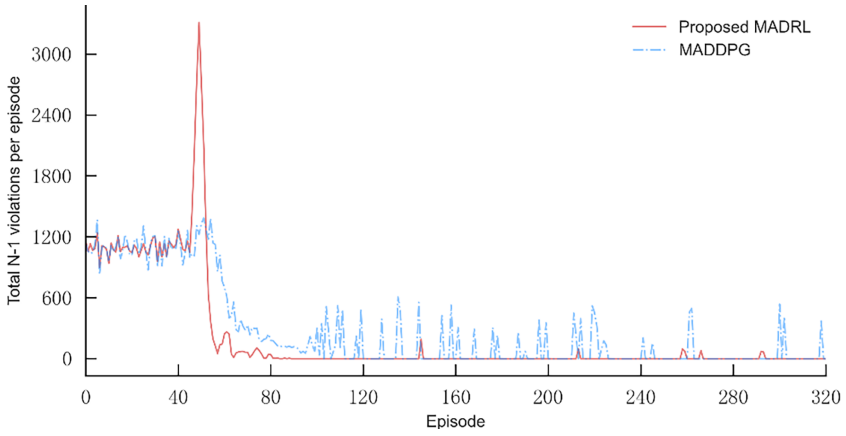


Figure 9: Algorithm comparison of cumulative N-1 violations during sequential load adjustments

Table 5: Comparison of different algorithms for control

Control strategy	Success rate	Decision time/ms	Action total number of times
MADDPG	96.20%	2.20	3882
Proposed method	99.20%	2.16	3770

5 Discussion

The proposed MADRL-based preventive control framework can be deployed within modern power system operation architectures as an intelligent auxiliary control layer. In practice, the trained agents can operate at the scheduling or regional control level, providing rapid preventive control recommendations based on real-time grid measurements. Because agent inference involves only lightweight matrix operations, the response time is within milliseconds, satisfying the timeliness requirement for short-term preventive control.

In large-scale power grids, communication delays and data acquisition latency may affect decision timeliness. These issues can be mitigated through a distributed control architecture in which regional agents operate independently but coordinate through shared state variables. This hierarchical structure aligns with the existing multi-level dispatching system in power networks. Future work will focus on integrating the MADRL framework with real-time measurement data (e.g., PMU/EMS) and improving robustness against communication uncertainty and data delays.

6 Conclusion

This paper proposes a curriculum-guided multi-agent deep reinforcement learning framework for static security preventive control of power systems. The method leverages centralized training with distributed execution to enable model-free, online safety regulation. By introducing CL, the instability issues of conventional MADDPG in complex grid scenarios are mitigated, significantly enhancing training robustness. Simulation results on the IEEE 39-bus system demonstrate that the proposed approach can effectively improve the static security compliance rate under N-1 contingencies and maintain stable performance under source–load fluctuations. These findings highlight its potential for practical application in distributed, real-time

decision-making. Future research will consider extending the framework to larger power system models and incorporating multiple types of uncertainties.

Acknowledgement: None.

Funding Statement: This research was funded by the China Southern Power Grid Co., Ltd. "Key technologies for stability analysis and coordinated control of new power systems based on data-mechanism fusion" (ZBKJXM20232027).

Author Contributions: The authors confirm contribution to the paper as follows: Ximing Zhang conceived the study, designed the methodology, conducted simulations, and managed the project. Zhuohuan Li was responsible for data analysis, curation, visualization, and writing the original draft. Xuexia Quan contributed to validation, reviewing, and editing. Kai Cheng provided critical resources and assisted with formal analysis. Yang Yu contributed to investigation and data collection. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article. And the additional data that support the findings of this study are available on request from the corresponding author, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Alizadeh MI, Usman M, Capitanescu F. Envisioning security control in renewable dominated power systems through stochastic multi-period AC security constrained optimal power flow. Int J Electr Power Energy Syst. 2022;139(2):107992. doi:10.1016/j.ijepes.2022.107992.
2. Capitanescu F, Martinez Ramos JL, Panciatici P, Kirschen D, Marano Marcolini A, Platbrood L, et al. State-of-the-art, challenges, and future trends in security constrained optimal power flow. Electr Power Syst Res. 2011;81(8):1731–41. doi:10.1016/j.epsr.2011.04.003.
3. Deng J, Wu G, Wang Y, Su Y, Liu A. Security-constrained hybrid optimal energy flow model of multi-energy system considering N-1 component failure. J Energy Storage. 2023;64(1):107060. doi:10.1016/j.est.2023.107060.
4. Wang C, Feng C, Zeng Y, Zhang F. Improved correction strategy for power flow control based on multi-machine sensitivity analysis. IEEE Access. 2020;8:82391–403. doi:10.1109/ACCESS.2020.2989927.
5. Sennewald T, Linke F, Westermann D. Preventive and curative actions by meshed bipolar HVDC-overlay-systems. IEEE Trans Power Deliv. 2020;35(6):2928–36. doi:10.1109/TPWRD.2020.3011733.
6. Poyrazoglu G, Oh H. Optimal topology control with physical power flow constraints and N-1 contingency criterion. IEEE Trans Power Syst. 2015;30(6):3063–71. doi:10.1109/TPWRS.2014.2379112.
7. Wang Z, Hao Y, Zhao G, Wang M, Su Z, Zhang J. Day-ahead optimal dispatch method of integrated electric-heat-cool-gas energy system based on N-1 safety criterion. Energy Build. 2024;323(3201–5):114800. doi:10.1016/j.enbuild.2024.114800.
8. Heidarifar M, Andrianesis P, Ruiz P, Caramanis MC, Paschalidis IC. An optimal transmission line switching and bus splitting heuristic incorporating AC and N-1 contingency constraints. Int J Electr Power Energy Syst. 2021;133(3):107278. doi:10.1016/j.ijepes.2021.107278.
9. Zhou F, Li L, Jia T, Yin Y, Shi A, Xu S. Intelligent power grid load transferring based on safe action-correction reinforcement learning. Energy Eng. 2024;121(6):1697–711. doi:10.32604/ee.2024.047680.
10. Wang L, Wang X. Enhanced deep reinforcement learning strategy for energy management in plug-in hybrid electric vehicles with entropy regularization and prioritized experience replay. Energy Eng. 2024;121(12):3953–79. doi:10.32604/ee.2024.056705.

11. Ding J, Chen B, Lei Y, Zhang W. Coordinated scheduling of electric-hydrogen-heat trigeneration system for low-carbon building based on improved reinforcement learning. Energy Eng. 2025;122(11):4561–77. doi:10.32604/ee.2025.067574.
12. Zhang H, Sun X, Lee MH, Moon J. Deep reinforcement learning-based active network management and emergency load-shedding control for power systems. IEEE Trans Smart Grid. 2024;15(2):1423–37. doi:10.1109/TSG.2023.3302846.
13. Chen Y, Zhu J, Liu Y, Zhang L, Zhou J. Distributed hierarchical deep reinforcement learning for large-scale grid emergency control. IEEE Trans Power Syst. 2024;39(2):4446–58. doi:10.1109/TPWRS.2023.3298486.
14. Xie J, Sun W. Distributional deep reinforcement learning-based emergency frequency control. IEEE Trans Power Syst. 2022;37(4):2720–30. doi:10.1109/TPWRS.2021.3130413.
15. Sun LJ, Gu XP, Liu T, Wang TQ, Yang XD. Active power security correction method for power grids based on deep reinforcement learning algorithm. Power Syst Prot Control. 2022;50:114–22 (In Chinese).
16. Wang T, Tang Y, Huang Y, Chen X, Zhang S, Huang H. Automatic adjustment method of power flow calculation convergence for large-scale power grid based on knowledge experience and deep reinforcement learning. In: Proceedings of the 2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2); 2020 Oct 30–Nov 1; Wuhan, China. p. 694–9. doi:10.1109/ei250167.2020.9346831.
17. Zeng H, Zhou Y, Guo Q, Cai Z, Sun H. Distributed deep reinforcement learning-based approach for fast preventive control considering transient stability constraints. CSEE J Power Energy Syst. 2021;9(1):197–208. doi:10.17775/CSEEJPES.2020.04610.
18. Li BY, Zhao JM, Han XQ, Yang J. Static security preventive control method of power systems based on dual-agent deep reinforcement learning. Proc CSEE. 2023;43:1818–30. (In Chinese). doi:10.13334/j.0258-8013.pcsee.220005.
19. Sun X, Qiu J. Two-stage volt/var control in active distribution networks with multi-agent deep reinforcement learning method. IEEE Trans Smart Grid. 2021;12(4):2903–12. doi:10.1109/TSG.2021.3052998.
20. Liu H, Zhang C, Chai Q, Meng K, Guo Q, Dong ZY. Robust regional coordination of inverter-based volt/var control via multi-agent deep reinforcement learning. IEEE Trans Smart Grid. 2021;12(6):5420–33. doi:10.1109/TSG.2021.3104139.
21. Wang S, Duan J, Shi D, Xu C, Li H, Diao R, et al. A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning. IEEE Trans Power Syst. 2020;35(6):4644–54. doi:10.1109/tpwrs.2020.2990179.
22. Chen X, Zhang M, Wu Z, Yu L, Hatziargyriou ND, Guan X. Load frequency control of multi-microgrids based on deep deterministic policy gradient integrated with online learning. IEEE Trans Smart Grid. 2025;16(5):4266–78. doi:10.1109/TSG.2025.3587312.
23. Yang Y, Li J, Hou J, Wang Y, Zhao H. A policy gradient algorithm to alleviate the multi-agent value overestimation problem in complex environments. Sensors. 2023;23(23):9520. doi:10.3390/s23239520.
24. Wang X, Chen Y, Zhu W. A survey on curriculum learning. IEEE Trans Pattern Anal Mach Intell. 2022;44(9):4555–76. doi:10.1109/TPAMI.2021.3069908.
25. Xiao T, Chen Y, Diao H, Huang S, Shen C. Fast-converging deep reinforcement learning for optimal dispatch of large-scale power systems under transient security constraints. J Mod Power Syst Clean Energy. 2024;13(5):1495–506. doi:10.35833/mpce.2024.000624.
26. Li H, Qian X, Song W. Prioritized experience replay based on dynamics priority. Sci Rep. 2024;14(1):6014. doi:10.1038/s41598-024-56673-3.