Efficient and Stable Learning for Distribution Network Operation: A Model-based Reinforcement Learning Approach

Dong Yan, Zhan Shi, Xinying Wang, *Senior Member*, *CSEE*, Yiying Gao, Tianjiao Pu, *Senior Member*, *CSEE*, and Jiye Wang, *Fellow*, *CSEE*

***Abstract*—This paper discusses the application of deep reinforcement learning (DRL) to the economic operation of power distribution networks, a complex system involving numerous flexible resources. Despite the improved control flexibility, traditional prediction-plus-optimization models struggle to adapt to rapidly shifting demands. Modern artificial intelligence (AI) methods, particularly DRL methods, promise faster decision-making but face challenges, including inefficient training and real-world application. This study introduces a reward evaluation system to assess the effectiveness of various strategies and proposes an enhanced algorithm based on the Model-based DRL approach. Incorporating a state transition model, the proposed algorithm augments data and enhances dynamic deduction, improving training efficiency. The effectiveness is demonstrated in various operational scenarios, showing notable enhancements in rationality and transfer generalization.**

***Index Terms*—Distribution networks, economic operation, reinforcement learning, reward shaping, transition model.**

I. Introduction

THE increasing integration of renewable energy (RE) sources and power electronic devices is transforming power systems towards smarter, more responsive, and more flexible smart grids. In light of this, developing synergies between sources, grids, loads, and storage has become an important research direction. Distribution networks (DN), as a vital component of power systems, are the primary integration where these multi-element sources, grids, loads, and storage aggregate. While the expandable control degrees of freedom from these elements enhance power systems' regulatory flexibility and proactiveness, they also impose heightened demands for real-time data processing and instantaneous, accurate responses [1].

To enhance the efficiency of DN optimization, decentralized multi-agent synergy of large-scale systems can be achieved through techniques based on the ADMM [2]. Combined with robust optimization techniques, even with information gaps and biases, system stability under complex scenarios can be ensured [3]. Additionally, stochastic optimization can obtain strategies with confidence levels by constructing deterministic optimization problems via scenario sampling [4] or chance constraint [5] methods. However, these methods often involve extremely large constraint sets or iterative verification of multiple subproblems, requiring simplified system models to ensure solving time. Moreover, the conservativeness of robust strategies constrains the response flexibility of resources.

In recent years, with the continuous advancement of artificial intelligence (AI), reinforcement learning (RL) techniques have been extensively explored and deployed across various power system applications [6], from power flow adjustment [7], voltage regulation [8], economic dispatch [9], [10] and demand efficiency improvements [11], [12]. Studies such as reference [13] propose a deep reinforcement learning (DRL) driven scheduling approach for energy systems, employing the Soft Actor-Critic (SAC) algorithm [14]. Reference [15] advances an adaptive uncertainty economic dispatch method for power systems, leveraging the Deep Deterministic Policy Gradient (DDPG) algorithm with differential information correction. Reference [16] introduces a Proximal Policy Optimization (PPO) based method to enhance collaboration among various energy types within local markets and to boost energy efficiency via customized responses. The aforementioned studies employ model-free DRL free from intricate dynamic modeling yet require the Markov property and stability of state transitions. Reference [17] introduces historical boundary data to supplement state modeling, employing Gated Recurrent Unit (GRU) deep neural networks (DNN) to extract temporal hidden features from feature matrices.

Furthermore, safety constraints, which serve as the primary boundaries in power system optimization, cannot be directly enforced through reward feedback. Some studies focus on safe RL methods to address these issues. Paper [18] proposes a DN optimization approach based on the Constrained Policy Optimization (CPO) algorithm, maintaining an independent evaluation DNN to assess the violation of system constraints. Reference [19] develops an online method for microgrid optimization based on the Lagrangian Soft Actor-Critic (LSAC)

Manuscript received November 14, 2023; revised January 5, 2024; accepted February 19, 2024. Date of online publication January 10, 2025; date of current version March 5, 2025. This work was supported by the National Key Research and Development Program of China (2022ZD0119804) and National Natural Science Foundation of China (Key Program), NSFC-SGCC (U2066213)

D. Yan (corresponding author, email: yandongloyid@126.com), Z. Shi, X.Y. Wang, Y. Y. Gao and T. J. Pu are with China Electric Power Research Institute, Beijing 100192, China.

J. Y. Wang is with State Grid Digital Technology Holding Co., Ltd., Beijing 100073, China.

DOI: 10.17775/CSEEJPES.2023.09100

algorithm, improving the SAC objective function through Lagrange multipliers and directly training the model via neural network. Similarly, reference [8] uses Lagrange to correct the voltage control reward function, ensuring the action limits. However, when violation instances are insufficient, the biases of constraint evaluation increase, negatively impacting policy training. Power system optimization can be modeled as an optimization problem whose solutions strictly satisfy the constraints. Adopted by reference [20], [21], neural networks are only used to compute storage behaviors while other devices are solved by optimal power flow (OPF) methods, significantly improving the reliability.

In recent years, novel methods such as model-based RL (MBRL) [20], offline RL, and meta-RL [22] have emerged aiming to address bottlenecks. Model-based (MB) methods represent an active research area, utilizing interaction samples for policy evaluation and improvement and maintaining state transition models constructed from parametric structures within the model-free learning architecture. Reference [21] proposes an MB method using MuZero for microgrid scheduling. Monte Carlo tree search (MCTS) describes state transition models to estimate future states and values, forming adaptable strategies to different source-load patterns. However, this method necessitates discretizing sources, loads, and outputs to fit the MCTS structure.

Building on relevant studies, this paper explores ways to improve the RL training efficiency and multi-scenario adaptability of DN optimization policy. Regarding decision modeling, OPF aids in constructing the RL interaction mechanism, and the reward-shaping method constructs comparative feedback to reduce the impacts of uncontrollable transitions. Modifications to the Model-Based Policy Optimization (MBPO) algorithm [23] are proposed to accommodate DN optimization characteristics and leverage the data efficiency advantages of MBRL, thereby improving policy training efficiency. The RL agent is trained and tested using real-world data to validate optimization performance and generalization improvements in a DN simulation environment.

II. Modeling of Distribution Network

The following elaborates on the economic dispatch model of the DN that is cogent to the described algorithm. This model can generate optimal theoretical solutions within predefined constraints, serving dual purposes. Firstly, it forms the basis for the original problem targeted at optimization solutions. Secondly, it lays the groundwork for constructing RL models, including the state, action, reward, and transition models. This forms a DN source-grid-load-storage synergy system that flexibly adjusts power based on the electricity market, demand fluctuations, and RE output variability. The objective of the system considers numerous cost factors, including the operating costs of distributed generators (DG), storage depletion, load demand response (DR), and curtailed electricity. To be consistent with the actual operation, costs associated with the purchase from the external and the grid loss are also considered in the objective. It can be expressed in (1)–(6).

$$\min \mathbb{E}\Bigg\{ \sum_{t=\Delta t}^{\mathrm{T}} \Bigg(\sum_{i\in\mathcal{G}} C_i^{\mathrm{DG}}(P_i^{\mathrm{DG}}(t)) + \sum_{j\in\mathcal{S}} C_j^{S}(P_j^{S}(t)) + \sum_{k\in\mathcal{D}} C_k^{\mathrm{DR}}(P_k^{\mathrm{DR}}(t)) + \sum_{n\in\mathcal{R}} C_n^{\mathrm{Cur}}(P_n^{\mathrm{Cur}}(t)) + C^{\mathrm{Ext}}(P^{\mathrm{Ext}}(t)) + C^{\mathrm{GL}}(t) \Bigg) \Bigg\} \tag{1}$$

$$C_i^{\mathrm{DG}}(P_i^{\mathrm{DG}}(t)) = (\alpha_i (P_i^{\mathrm{DG}}(t))^2 + \beta_i P_i^{\mathrm{DG}}(t) + c_i)\Delta t \tag{2}$$

$$C_j^{S}(P_j^{S}(t)) = \sigma_j |\mathrm{SoC}_j(t) - \mathrm{SoC}_j(t-\Delta t)| \tag{3}$$

$$C_k^{\mathrm{DR}}(P_k^{\mathrm{DR}}(t)) = \zeta_k P_k^{\mathrm{DR}}(t)\Delta t \tag{4}$$

$$C_n^{\mathrm{Cur}}(P_n^{\mathrm{Cur}}(t)) = \rho_n P_n^{\mathrm{Cur}}(t)\Delta t \tag{5}$$

$$C^{\mathrm{Ext}}(P^{\mathrm{Ext}}(t)) = \lambda^{\mathrm{buy}}(t) P^{\mathrm{Ext}}(t)\Delta t \tag{6}$$

In Equations, C symbolizes the cost functions. The power of respective devices is represented by P. The set of node numbers where the devices are located is denoted by $\mathcal{G}, \mathcal{S}, \mathcal{D}, \mathcal{R}$, while i, j, k, n signifies the elements of these sets. The operating costs of DGs are modeled using a quadratic function, where α, β, c represents the quadratic, linear, and constant coefficients. The storage depletion costs, represented by C^S, are calculated using the state of charge (SoC) and the depletion coefficient σ. The costs of DR, curtailment, and external grid interaction are modeled linearly, with ζ, ρ, λ signifying the cost coefficients. The objective is to minimize cumulative operating costs, where the total duration, the current moment, and the optimization resolution are encapsulated by $T, t, \Delta t$. The establishment of an optimization model requires stringent support constraints. In this specific problem, constraints are modeled individually for each controllable object. The detailed forms of these constraints can be found in the Appendix A. The energy balance is established under the Convex Relaxation Branch Flow Model [24]. Under this setting, the network loss can also be expressed as below.

$$C^{\mathrm{GL}}(t) = \eta^{\mathrm{GL}} \sum_{(i,j)} r_{ij} l_{ij} \tag{7}$$

where η^{GL} represents the target coefficient, l_{ij} represents the square of the branch current modulus and r_{ij} represent the branch resistance.

III. Modeling of Decision Process

A. Modified MDP for Distribution Network Optimization

Applying RL to the optimization of DN necessitates the transformation of optimization models into MDP frameworks. The decision-making inherent in DN optimization typically involves features such as the boundary at the present timestep and the real-time variables. The boundary, generally determined by the environment, signifies the non-controllable states, while the real-time variables, modifiable through strategic planning, can be viewed as a deterministic transition process.

Traditional MDP for DN optimization lacks a clear criterion for the success of tasks, and the state space includes numerous critical features that are irrelevant to the policy but have a significant impact on the reward value. Moreover, these rewards are affected by uncontrollable states, which means

the performance of the policy is highly correlated with the boundary conditions of the scenarios, which demands higher stability in state transitions. The traditional MDP model needs to be modified to reduce the algorithm's dependence on deterministic scenarios. After distinguishing the controllability of elements in the state space s, the transition process of the decision can be expressed as shown in Fig. 1.

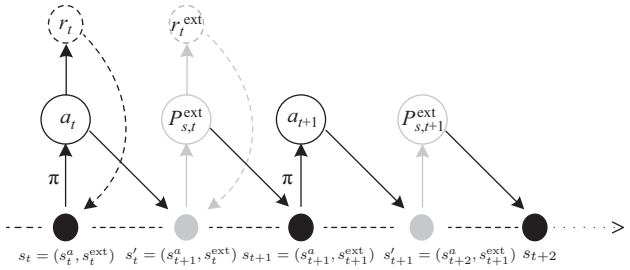

Fig. 1. Modified MDP Process.

In this figure, s_t, s_t^a and s_t^{ext} denote the state space, the state tensor explicitly impacted by the action space, and the environmental state feature unaffected by the policy. Furthermore, π symbolizes the policy, and a_t represents the action derived from the current state and corresponding policy. The reward feedback, obtained after action execution, is portrayed by r_t. Affected by the policy, s_t deterministically transitions s'_t. Subsequently, s'_t shifts to s_{t+1} under the influence of the environmental transition $P_{s,t}^{\text{ext}}$, whilst acquiring external environmental change feedback on the evaluation function policy, denoted by r_t^{ext}.

In discussing the aforementioned Markov reward model, since the transition feedback must be constructed using the after state, reward function can be represented as (8)–(9).

$$r_t = r'(s_t, a_t, s_{t+1}^a) = r'(s_t, a_t) \tag{8}$$

$$r_t^{\text{ext}} = r''(s_t^{\text{ext}}, a_t, s_{t+1}^{\text{ext}}) \tag{9}$$

The evaluation function r_t^{ext} quantifies the compatibility between the policy and environment. Since the influence of s_t^{ext} is only effective at the current time step, if its contribution outweighs the policy, the influence will reverberate to previous step via discount factor. Assuming that the reward determined by s_t^{ext} can be explicitly independent, then r_t can be further decomposed into the (10).

$$r_t = r_t^{\pi}(s_t, a_t) + r_t^{\text{env}}(s_t^{\text{ext}}) \tag{10}$$

Define r^{env} as the environmental baseline (EB), primarily determining the relative offset of the reward value domain. It represents the theoretical lower limit of immediate benefits at time t. Define r^{π} as the policy benefits (PB), indicating the theoretical range of reward changes achievable by adjusting the policy. Given the assumption of the form (10), the expression of its state-action value function Q is obtained, as shown in (11). The Q value, initially undefined, is determined by model dynamics transition and policy distributions, symbolized as $s_{t+1} \sim f(s_t^a, a_t) \cdot P_{s,t}^{\text{ext}}, a_t \sim \pi(s_t)$.

$$\begin{aligned} Q_\pi(s_\tau, a_\tau) &= \mathbb{E}_\pi \sum_{k=0}^{\infty} \gamma^k r(s_{\tau+k}, a_{\tau+k}) \\ &= \mathbb{E}_\pi \sum_{k=0}^{\infty} \gamma^k r^{\pi}(s_{\tau+k}, a_{\tau+k}) + \mathbb{E}_\pi \sum_{k=0}^{\infty} \gamma^k r^{\text{env}}(s_{\tau+k}^{\text{ext}}) \\ &= \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r^{\pi}(s_{\tau+k}, a_{\tau+k}) \right] + \underbrace{\frac{1}{M} \sum_{i=0}^{M} \sum_{k=0}^{\infty} \gamma^k r^{\text{env}}(s_{\tau+k,i}^{\text{ext}})}_{Q_b} \end{aligned} \tag{11}$$

The (11) indicates that the transition of s^{ext} impacts the policy value function Q_π in three distinct manners. Firstly, it influences the expected policy returns via the transition probabilities. Secondly, it indirectly modifies the magnitude of policy returns by affecting the value range of r^{π}. Thirdly, it introduces an extraneous term Q_b unrelated to the policy. Upon examining Q_b, it becomes apparent that the coefficient γ^k decays over time, suggesting that the value of the additional term is influenced by the chronological ordering of s^{ext}. When the policy undergoes training within a novel scenario, the value function expectation derived from the historical trajectory must offer a biased estimate regardless of the sampling method.

To counteract the inherent biases embedded within the evaluation system, especially the issue where r^{π} lack a consistent comparability under various s^{ext}, as well as the challenge of rectifying the additional terms Q_b, subsequent sections will introduce modeling approaches grounded in RL problems.

B. Problem Reformulation for Reinforcement Learning

Based on the economic dispatch model of DN, we initially construct a feature set for state observation. The state space at time t, comprising components s_t^a and s_t^{ext}, should each exhibit Markovian state transition properties. For s_t^a, given the states of various dispatch elements at time t following the decision at t-1, it can be articulated as (12).

$$s_t^a = \Big(\underbrace{P^{\text{DG}}(t-1), Q^{\text{DG}}(t-1)}_{i}, \underbrace{P^{\text{DR}}(t-1)}_{k}, \underbrace{\text{SoC}(t-1)}_{j} \Big) \tag{12}$$

For component S_t^{ext}, historical data are utilized to construct a two-dimensional feature tensor, as expressed through (13):

$$\begin{aligned} s_t^{\text{ext}} &= (o_{t-\tau}, \cdots o_t)^{\text{T}} \\ o_t &= (P_t^{L,1}, \cdots P_t^{L,k}, Q_t^{L,1}, \cdots Q_t^{L,k}, P_t^{pv,1}, \cdots P_t^{pv,m}, \\ &\quad P_t^{wt,1}, \cdots P_t^{wt,n}, \lambda_t^{\text{buy}}, \lambda_t^{\text{sell}}) \end{aligned} \tag{13}$$

A partition merging approach is employed to process historical information to streamline the state space features, using the sum of powers in each area as input. The reward function is typically constructed based on the cost function, assuming the form of either a negative or an exponential. Using the negative cost form as an example, the reward for a single-step decision is represented by (14), where $\varphi(t)$ denotes the compensation for constraints.

$$\begin{aligned} r_t = &\sum_{i \in \mathcal{G}} C_i^{\text{DG}}(P_i^{\text{DG}}(t)) + \sum_{j \in \mathcal{S}} C_j^{S}(P_j^{S}(t)) \\ &+ \sum_{k \in \mathcal{D}} C_k^{\text{DR}}(P_k^{\text{DR}}(t)) + \sum_{n \in \mathcal{R}} C_n^{\text{Cur}}(P_n^{\text{Cur}}(t)) \\ &+ C^{\text{Ext}}(P^{\text{Ext}}(t)) + C^{\text{GL}}(t) + \varphi(t) \end{aligned} \tag{14}$$

The multi-time-step optimization problem can be converted into multiple single-step optimizations through pertinent temporal constraint decoupling. This facilitates RL iteration, implying that the OPF method can replace the power flow, which gives feedback on the policy. Only the temporally strongly coupled variables are decided by RL agents, while other temporally unrelated or weakly coupled variables are obtained from OPF. Additionally, OPF ensures the policy safety boundary, which means $\varphi(t)$ is always 0.

In practical problems, the DR and RE exhibit weak temporal coupling due to their flexibility. Moreover, the DG constraints are treated as shown in the Appendix (A1) can achieve temporal decoupling. Here, $P^{\mathrm{DG}*}$ represents the DG actions at time $t-1$.

$$\begin{cases} P_i^{\mathrm{DG}} \le P_i^{\mathrm{DG}}(t) \le \bar{P}_i^{\mathrm{DG}} \\ P_i^{\mathrm{DG}*}(t-1) - \eta_i^{\mathrm{DG}} \bar{P}_i^{\mathrm{DG}} \le P_i^{\mathrm{DG}}(t) \\ \quad \le P_i^{\mathrm{DG}*}(t-1) + \eta_i^{\mathrm{DG}} \bar{P}_i^{\mathrm{DG}} \end{cases} \tag{15}$$

While the output of storage devices is directly affected by SOC, their controllability exhibits strong temporal coupling. Therefore, it is retained in the agents' action space. Thus, the objective of the OPF can be expressed in (16).

$$\begin{aligned} \min \Bigg(& \sum_{i\in\mathcal{G}} C_i^{\mathrm{DG}}\left(P_i^{\mathrm{DG}}(t)\right) + \sum_{k\in\mathcal{D}} C_k^{\mathrm{DR}}\left(P_k^{\mathrm{DR}}(t)\right) \\ & + \sum_{j\in\mathcal{S}} C_j^{S}\left(P_j^{S}(t), \mathrm{SoC}(t-1)^*\right) + C^{\mathrm{Ext}}(P^{\mathrm{Ext}}(t)) \\ & + \sum_{n\in\mathcal{R}} C_n^{\mathrm{Cur}}\left(P_n^{\mathrm{Cur}}(t)\right) + C^{\mathrm{GL}}(t) | P_j^S(t) = P_j^{S*}(t) \Bigg) \end{aligned} \tag{16}$$

The target (16) employs $\mathrm{SoC}^*(t-1)$, the SOC from the previous decision, to assess the storage cost. To ensure P^{S*} complies with constraints Appendix (A4)–(A5), when SOC reaches the upper bound and the decision is charging, or when SOC reaches the lower bound and the decision is discharging, P^{S*} takes the value 0. When the charging/discharging amount is out of the limits, P^{S*} takes the maximum power corresponding to the remaining capacity.

C. The Induction Function for Policy Benefits

The PB value describes the incremental reward relative to the EB, which is used to build the true return of policy. Due to the reward interval and change rate being jointly affected by s^{ext} and s^a, suggesting that the PB determined by each decision cannot be straightforwardly compared. In other words, a lower PB value at a certain step does not imply the agent getting an inferior state. Assuming homogeneity across operating scenarios, this paper posits that the reward value used to evaluate the policy should remain consistent. Utilizing the Modified MDP established previously, an induced mechanism based on r_t^{ext} is defined, compensating the PB value under differing external state features s^{ext}. In the context of the OPF-based interaction, where only storage is variable, its inducing function can be simplified as shown in (17).

$$\begin{aligned} r^{\mathrm{ext}}(t) &= \sum_{n_r} P_{n_r,t}^S \cdot \lambda_t^{\mathrm{buy}} \cdot \lambda_t^L / \lambda_t^R \\ \lambda_t^L &= \sum_{j\in n_r} (P_{j,t}^L + P_{j,t+1}^L) / \bar{P}_{n_r}^L \\ \lambda_t^R &= \sum_{j\in n_r} (P_{j,t}^{wt} + P_{j,t}^{pv} + P_{j,t+1}^{wt} + P_{j,t+1}^{pv}) / (\bar{P}_{n_r}^{wt} + \bar{P}_{n_r}^{pv}) \end{aligned} \tag{17}$$

Equation (17), n_r denotes power grid zones, while $\lambda_t^L, \lambda_t^{\mathrm{RE}}$ symbolizes the relative level of grid loads and RE outputs, indicative of the system state migration's direction. The reward articulated by the induction aligns with the system optimization trend, incentivizing storage to actively charge when λ_t^{RE} is higher, λ_t^{RE} is lower and price level is lower. However, charging behavior corresponds to an actual increase in system cost, so the immediate feedback term in the induction function is negative, reducing its absolute value to encourage charging behavior at suitable times. The opposite is also true. The induced reward r^{ext} is summed with the original reward r_t to dilute the effect of s^{ext}, which partially restores the superiority and inferiority expressed by the state value.

D. The Correction for Environmental Baseline

Unlike the indirect impact on PB, the additional term Q_b in function (11) directly affects the value stability. Ideally, the true PB value can be obtained by subtracting rewards r^{env}. However, directly calculating r^{env} poses difficulties. Firstly, solving a max-min problem is time consuming, affecting RL's exploration efficiency over multiple rounds. Secondly, the relaxed branching flow constraints with the unclosed upper bound make it challenging to ensure the feasibility of the maximization.

Considering that the feedback from any legal action must contain the component r^{env}, a logical approach is to establish a policy baseline and use the corresponding step reward values a benchmark. Given that the aforementioned induction modification pertains solely to the action taken, ensuring that the EB correction does not alter its value is imperative. Defining the baseline policy as the collection of actions whose induction value is zero, which can be expressed in (18).

$$\begin{aligned} & r(s_t, \hat{a}_t, s_{t+1}) \\ & = r^{\pi}(s_t, \hat{a}_t) + r^{\mathrm{env}}(s_t^{\mathrm{ext}}) + r^{\mathrm{ext}}(s_t, \hat{a}_t, s_{t+1}) \\ & = r^{\pi}(s_t, \hat{a}_t) + r^{\mathrm{env}}(s_t^{\mathrm{ext}}) + 0 \to r^{\mathrm{BL}}(s_t, \hat{a}_t) \end{aligned} \tag{18}$$

Then the reward feedback with the baseline correction can be expressed in the form (19).

$$\begin{aligned} & r(s_t, a_t, s_{t+1}) - r^{\mathrm{BL}}(s_t, \hat{a}) \\ & = r^{\pi}(s_{t+k}, a_{t+k}) - r^{\pi}(s_{t+k}, \hat{a}) + r^{\mathrm{ext}}(s_t, a_t, s_{t+1}) \end{aligned} \tag{19}$$

Zero induction leads to zero storage actions under OPF based environment, and the corresponding baseline reward equals to the optimal cost when the storage does not participate in system operation. This avoids the coupling and association of SOC over time sequences on the one hand, and converts the max-min optimization into a single minimization problem on the other hand, which is computationally simpler to resolve.

IV. Model-Based Policy Optimization Algorithm Based on Modified MDP

A. Algorithm Framework

To approximate the transition probabilities accurately, a more robust and diverse environmental feature set is necessary for policy training. However, noting that every training step necessitates OPF computations is crucial. Exploring varying actions under comparable scenarios implies the repeated resolution of complex, approximate models. This highlights an inherent issue pertaining to the efficiency of data utilization. MBRL methodologies construct a dynamic model that facilitates learning state transitions and rewards. Furthermore, it provides an expanded scope of possible scenarios under the assumption of stable transitions.

MBPO, an MBRL method within the Dyna Q framework [25], constructs a system dynamic model as shown in (20) to enable state trajectory prediction. It then integrates this with other model-free RL methods for policy updating.

$$s_t^k, r^k = f_\theta(s_t^{k-1}, a_\phi) \tag{20}$$

k is the step number for generating a trajectory from the real state, and θ, ϕ denotes the network parameters of the dynamic model and the policy. Based on this paradigm, numerous enhancements to MBRL algorithms have been proposed, including Muzero and Dreamer [26], which have achieved breakthroughs on classic RL problems.

B. Analysis of Return Bound

To analyze the capability of policy improvement, it is typically necessary to construct a return boundary relationship as shown in (21). Here, $\eta(\pi)$ represents the return of the real decision-making process, while $\hat{\eta}(\pi)$ signifies the decision made through model rollouts. C is a constant that represents the gap between two processes. We aim to ensure that each policy update results in a return improvement not less than C. The smaller the C is, the easier to get the improvement.

$$\eta(\pi) = \hat{\eta}(\pi) - C \tag{21}$$

Theorem 1: Assume that the maximum expected KL divergence between the dynamic model and the real state transition is ε_m, and the maximum expected KL divergence of distribution bias during the update process is ε_π, which are shown in (22)–(23), the return bound corresponding to the Modified MDP for DN optimization satisfies (24)–(25).

$$\varepsilon_m \geq \max_t E_{s\sim\pi_{D,t}}[D_{KL}(p(s',r|s,a)||p_\theta(s',r|s,a))] \tag{22}$$

$$\varepsilon_\pi \geq \max_s D_{\mathrm{KL}}(\pi||\pi_D) \tag{23}$$

$$\eta[\pi] \geq \hat{\eta}[\pi] - 2r_{\max}\gamma\frac{\varepsilon_m + 2\varepsilon_\pi}{(1-\gamma)^2} \tag{24}$$

$$r_{\max} = \max|r^{\mathrm{env}}(s^{\mathrm{ext}})| + \max|r^{\mathrm{ext}}(s_t, a_t, s_{t+1})| \tag{25}$$

Proof. See Appendix B.

Based on the (24)–(25), consider reducing the bound further. First discuss the changes in the coefficient $r_{\max}$ after introducing the baseline correction. The upper bound of the absolute value after introducing r^{BL} is shown in the following (26).

$$\begin{aligned}&|r'(s,a,s')|\\ &= |r^\pi(s_{t+k}, a_{t+k}) - r^\pi(s_{t+k}, \hat{a}) + r^{\mathrm{ext}}(s_t, a_t, s_{t+1})|\\ &\leq |r^\pi(s_{t+k}, a_{t+k}) - r^\pi(s_{t+k}, \hat{a})| + |r_t^{\mathrm{ext}}(s_t, a_t, s_{t+1})|\end{aligned} \tag{26}$$

Compared with formula (25), the first item changes from the upper bound of r^{env} to the maximum difference between the r^π of strategy and its baseline, which can easily get a basic relationship can be derived as (27).

$$\begin{aligned}|r^\pi(s_t, a_t) - r^\pi(s_t, \hat{a})| &\leq \max_{s,a}|r^\pi(s,a)|\\ &< \max_s |r^{\mathrm{env}}(s^{\mathrm{ext}})|\end{aligned} \tag{27}$$

In conclusion, under the same reward settings, baseline correction can reduce the bias, making it easier for the policy update to meet the difference limit requirements.

In addition, formula (24) depends entirely on the interaction with the rollout model. The MBPO algorithm proposes a k-step rollout method, which inserts a rollout branch in the exploration and predicts the trajectory in the k-steps based on the actual state. Under this scheme, the sum of error will be split into two parts: the k steps of rollout in the learned model and the interaction of $t-k$ steps under the true dynamic.

Theorem 2: Assume that ε_m and ε_π are constrained by (22)–(23). Considering a k step rollout policy, its return bound can be represented as the following:

$$\eta(\pi) \geq \eta^{\mathrm{branch}}(\pi) - 2r_{\max}\left[\frac{\gamma^k}{(1-\gamma)^2}\varepsilon_\pi + \frac{k+1}{1-\gamma}(\varepsilon_m + 2\varepsilon_\pi)\right] \tag{28}$$

where $\eta^{\mathrm{branch}}(\pi)$ represents the return of the policy using the k-step branch. For detailed derivations, please refer to Lemma A2, A3, and B4 in reference [22]. Theorem 2 shows that the first term of the bound decreases exponentially with k, while the second term increases linearly with k. Therefore, the k that minimizes the bound C is determined by the specific value of $\varepsilon_m, \varepsilon_\pi$. This will be further discussed in case study.

C. Training of Dynamic Model

The training of the Dynamic model is performed using samples collected through exploration. As the temporal state is a cyclic structure within a specific time window, calculating the variation amounts of only the cyclic replacement parts can further construct complete information. The complete label construction process is shown in Fig. 2. A mean squared error (MSE) loss function of the form (29) is used to train the Dynamic model. An ensemble learning approach is used to reduce the impact of parameter differences between models.

$$\begin{aligned}&\Delta\tilde{s}, \tilde{r} \sim \mathcal{N}(\mu_\theta, \sigma_\theta)\\ &J_\theta = \mathbb{E}\left[\sum_i (\Delta\tilde{s}_i - \Delta s_i)^2 + (\tilde{r} - r)^2\right]\end{aligned} \tag{29}$$

The rollout length, denoted as k, is articulated to gradually increase with the progression of training steps. Furthermore,

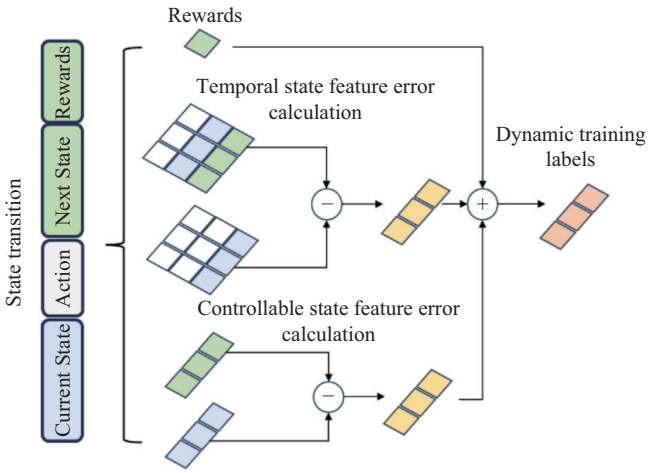

Fig. 2. Training label structures of dynamic model.

the actual k value during training is determined based on a predefined threshold value, as depicted in (30).

$$k = \lfloor \bar{k} \cdot n_{\text{step}} / N_{\bar{k}} \rfloor, \ n_{\text{step}} \leq N_{\bar{k}} \tag{30}$$

In the given equation, $\bar{k}$ signifies the upper boundary for step values. $n_{\text{step}}, N_{\bar{k}}$ corresponds to the present training step and the step threshold when k attains its maximum boundary. Moreover, $\lfloor \cdot \rfloor$ is the floor function. Once n_{step} surpasses the specified threshold, k is constantly maintained at its upper limit.

D. Training of Policy Model

The MBPO leverages a model-free RL agent, which capitalizes on a combination of real and simulated samples to update policy. Drawing parallels with the foundational method, the algorithm investigated in this study incorporates the SAC approach as a cornerstone for policy enhancement. As a maximum entropy RL methodology, SAC ensures a more robust training process. The objective specifically employed to upgrade the policy network is delineated in (31).

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim \mathcal{D}} \left[D_{\text{KL}} \left(\pi_\phi(\cdot|s_t) \| \frac{\exp(Q_\varphi(s_t, \cdot))}{Z(s_t)} \right) \right] \tag{31}$$

In this equation, the terms ϕ, φ are representative of the parameters for the policy and critic networks, respectively, with Z serving as the normalization coefficient. The objective function deployed to train the Critic network is illustrated in (32)–(34).

$$J_V(\psi) = \mathbb{E}_{s_t \sim \mathcal{D}} \frac{1}{2} [V_\psi(s_t) - \mathbb{E}_{a_t \sim \pi_\phi} (Q_\varphi(s_t, a_t) - \log \pi_\phi(a_t|s_t))]^2 \tag{32}$$

$$J_Q(\varphi) = \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}} \left[\frac{1}{2} (Q_\varphi(s_t, a_t) - \hat{Q}(s_t, a_t))^2 \right] \tag{33}$$

$$\hat{Q}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim p} [V_{\bar{\psi}}(s_{t+1})] \tag{34}$$

In this representation, $\psi, \bar{\psi}$ signifies the parameters of the state value function network and their corresponding soft update parameters for the target network.

In conclusion, the comprehensive procedure for the whole MBPO algorithm, specifically designed for DN optimization, is delineated in Algorithm 1.

V. Numerical Analysis

A. Case Study Configurations

This paper improves upon IEEE123 standard case studies, referencing the resource allocation scheme from reference [15], which makes the optimizable objects consistent with the model described above. The additional element's detailed parameters are in the Appendix C.

The varied operating conditions are derived from publicly accessible data provided by CAISO [27], leveraging actual 15-minute time-series data to model electricity prices, load demand, and the maximum output of RE. To capture the inherent uncertainty associated with the output of REs, the dataset for renewables exhibits episodes of extreme low-wind events or substantial variability in power generation. Electricity prices are established within a range of 20–60. They serve as dimensionless indicators, providing a reasonable reflection of the economic operation of the system. For detailed information on the pattern data, please refer to Fig. D1.

B. Configuration for DNNs

This section elucidates the architecture of the neural networks employed for policy training. For the time-series matrix, which is composed of uncontrollable states, we utilize a Conv1d network to build sequential relationships. Contrarily, a fully connected network is implemented to encode the vector for controllable states and action values. Drawing from these representation networks, we construct branch-specific networks tailored to individual functions, as depicted in Fig. 3.

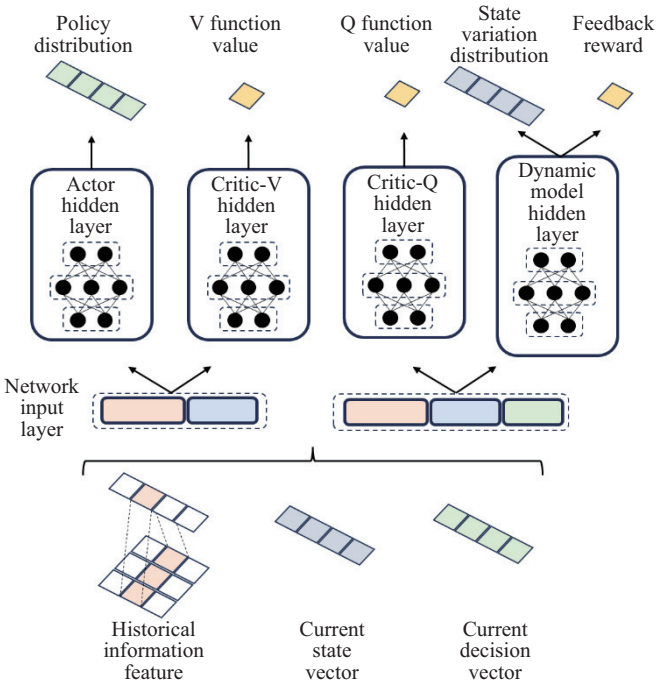

Fig. 3. DNN configuration for MBPO algorithm.

To reduce the input state space dimensionality for easier network training, the original standard case study is partitioned, by lines 13–18, 119–167, and 72–76, into 4 subregions. calculating the sum of outputs within each region and aggregating

Algorithm 1: The MBPO for distribution network optimization

Initialize Actor parameters ϕ, Critic parameters $\psi, \bar{\psi}, \varphi$, multiple predictive model network parameters θ_i, dataset D_{env} and D_{model}.
for *N epochs:* **do**
  Update the rollout length parameter k based on (30)
  for *E steps:* **do**
    Determine if the dynamic model parameters need to be updated.
    If True:
    Update predictive model parameters θ_i by SGD algorithm on D_{env}, according to (29).
    Sample the benchmark dataset from D_{env}, select the top n predict models with the smallest mean Loss value as elite rollout models.
    Calculate (s, a) on current state s based on current policy π_ϕ
    Substitute a into DN OPF model to solve the optimal objective value yields the immediate reward value r^{cost}.
    Calculate the induced reward value r^{ext} according to (17)
    Calculate the action a^{ind} and solve for corresponding r^{BL}.
    Calculate $r = r^{\text{cost}} + r^{\text{ext}} - r^{\text{BL}}$ and add samples (s_t, a_t, r, s_{t+1}) to D_{env}.
    for *M model rollouts:* **do**
      Sample s_t uniformly from D_{env}.
      for *k rollout step:* **do**
        Perform elite models starting from s_t and policy π_ϕ, add to D_{model}.
        $s_t = s_{t+1}^{\text{rollout}}$
      end
    end
    for *G gradient updates:* **do**
      Sampling strategy samples from D_{env} and D_{model} at a ratio γ_{rollout}.
      Update the state value Critic parameters ψ according to (32) by $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi, D_{\text{model}})$
      Update the Q Critic network parameters φ according to (33) by $\varphi \leftarrow \varphi - \lambda_Q \hat{\nabla}_\varphi J_Q(\varphi, D_{\text{model}})$
      Update the Actor network parameters ϕ according to (34) by $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi, D_{\text{model}})$
      Soft update target state value parameters $\bar{\psi}$ by $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$
    end
  end
end

data from the past 12 steps as historical information. Obtaining a historical feature matrix of dimension 17∗12. The Conv1d kernel for processing this feature is set to 4, stride of 1, and out channels of 5, yielding an expanded hidden feature dimension of 45 after Conv1d layer. Controllable states and action vectors are also expanded to 64-dimensional feature vectors via fully connected layers. The body of each independent functional network is structured as (512,256,256). For dynamic model training hyperparameters, the number of ensemble models is set to 5, the number of elite models to 3, the rollout step k ranges from 1 to 5, and each rollout generates 2560 samples. Samples generated account for 95% of the total batch size.

C. Results of Training and Testing

This section focuses on a comparative analysis of training results obtained through applying various DRL algorithms under the model configurations. To illustrate that MBPO can attain a policy enhancement equivalent to model-free RL algorithms. Still, with superior sample efficiency, it is juxtaposed with three of the most widely employed model-free RL methods: SAC, Twin Delayed Deep Deterministic Policy Gradient (TD3), and PPO. The reward function encompasses both induction and baseline corrections, and the training process utilizes scenarios constructed from the operational data of July, with each episode being constituted of 3000 steps. The aggregate rewards garnered by the algorithms are depicted in Fig. 4.

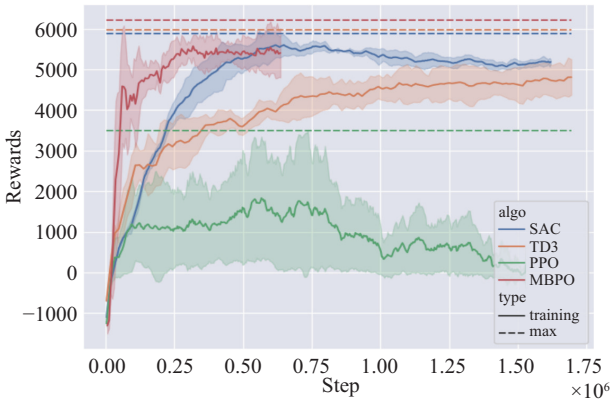

Fig. 4. Comparison of returns during the training process.

The dashed lines in the figure indicate the maximum reward. Modified MBPO achieved the highest value of 6714.38, while TD3 and SAC obtained similar values of 5987.93 and 5877.67

respectively. Examining the enhancement speed based on the average curves, it is clear that the modified MBPO can achieve the highest task return with the fewest exploration steps, requiring only around half as many steps as model-free RL. For other RL algorithms, TD3's reward raising is stable but slower; PPO even failed to learn effectively in this case.

In addition, the training performance variance caused by different random seeds, as shown in the figure shadows, is significantly more significant for modified MBPO despite its highest reward values. This is because random influences not only the Actor and Critic network updates, but also the quality of policies generated by the dynamic models, resulting in reduced parameter stability for the algorithm.

Figure 5 shows the testing result of applying the modified MBPO algorithm on the part of the case during periods of high RE generation-spanning 96 hours. Figure 5(a) presents the total load, total RE output, and dynamic prices over the verification period. Peak shaving and valley filling through battery participation are tested by calling the models, clearly showing RE output exceeds the total system load around the 25th and 48th hours.

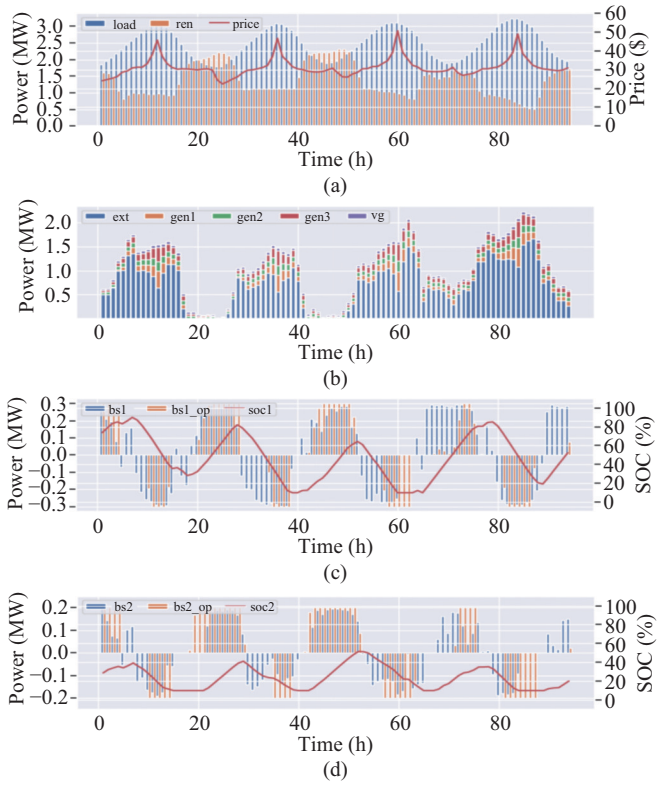

Fig. 5. Policy results on test cases including (a) boundary of load, RE output and price, (b) optimization results of DG output, DR output and external power interaction, (c) optimization results of storage output and SOC status at node 20, (d) optimization results of storage output and SOC status at node 56.

Comparing the agent decisions with the optimal solutions, the theoretical optimum always quickly adjusts the power to the limit and can maintain consistency in the strategy over a certain period. In contrast, the RL strategy will take energy storage actions before the optimum, and its power will rise and fall. This phenomenon is attributable to the lack of explicit information about the future, and the agent is willing to adopt such a conservative strategy.

D. Analysis of MDP Modifications

A series of comparative experiments with three reward configuration cases are studied to evaluate the effectiveness of the two-step reward modification proposed in the paper. Among them, Case 1 utilized a reward function that includes both EB and PB corrections. Case 2 employed a reward function that contained only the PB adjustment. Case 3 used only immediate cost reward function, without employing the methods proposed in the paper.

In Fig. 6, different color schemes represent different configuration cases, and line thickness is used to distinguish between the actual training and the reward recording. To reflect the modification effect under different algorithms, we applied SAC, TD3, and modified MBPO algorithms for case training. The optimal reward corresponding to Case 1 in each figure is superior to Case 2 and Case 3, demonstrating greater training stability under different random seed conditions. In terms of the rate of reward improvement, except for the early stages of TD3 training, the efficiency corresponding to Case 1 is also significantly better than the others, proving that corrections proposed in this paper can effectively improve the quality of training. Additionally, a progressive integration of the correction method starting from Case 3 leads to a steady increase in the accumulation of reward returns, suggesting that both correction methodologies are effective.

Comparing the second and third rows of the composite figure, it is known that the trends in reward changes before and after the addition of induced corrections are almost consistent, proving that the induced correction can serve as the representation of optimal policy value, which is consistent with the assumption mentioned above.

Comparing each column of the figure, the model-free algorithms, although the training without baseline corrections has some fluctuations, their returns can still be effectively improved. On the contrary, the MBPO algorithm corresponding to the third column will show a significant decline in returns, indicating the great impact of the improvement. This is mainly because the fluctuation of EB corresponding to uncontrollable states reduces the fitting accuracy of policy rewards, making it difficult for policy benefits to exceed the theoretical bound.

E. Comparison of Policy Generalization

To evaluate the generalizability of the agent, we selected several networks which yielded the best cumulative reward in various case settings. Each test environment was constructed based on different monthly operating data, with significant load, RE and price differences. To demonstrate the effect of the policy, the optimal solutions of the SOCP at storage withdrawal and the MISOCP with storage operation were calculated, representing the baseline policy performance and the theoretical optimal solution under certain operating conditions. Different agents' costs (COST) were calculated, and their relative improvement percentages (IMP) were derived. The training using both the induced and environmental correction is marked with the letter B, and the training using only the

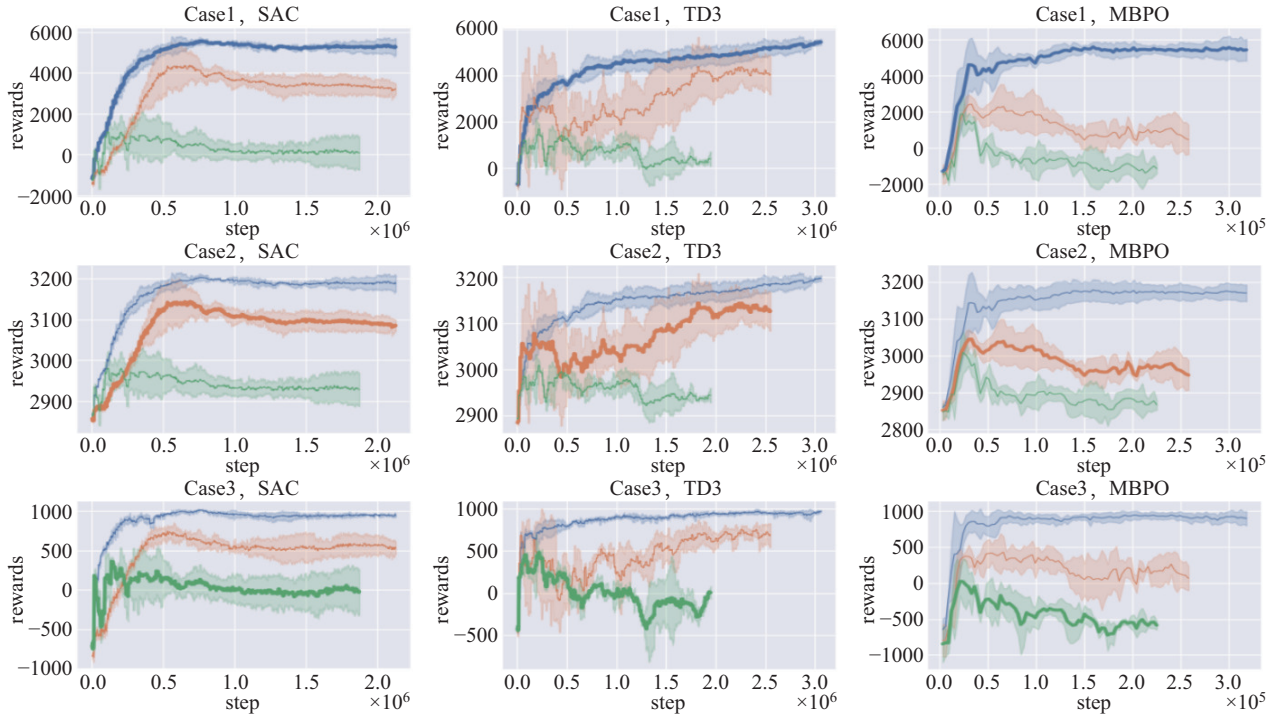

Fig. 6. Comparison of training process via diverse reward configuration.

TABLE I
TEST RESULTS OF DIFFERENT POLICY IN VARIOUS TEST CASES

	SOCP	MISOCP		MBPO-B		MBPO-I		SAC-B		SAC-I		TD3-B		TD3-I	
CASE	COST	COST	IMP	COST	IMP	COST	IMP	COST	IMP	COST	IMP	COST	IMP	COST	IMP
Jul.	2.874	2.692	6.35	2.693	6.32	2.737	4.79	2.697	6.18	2.707	5.83	2.697	6.16	2.728	5.10
Jan.	3.083	3.044	1.24	3.047	1.16	3.059	0.78	3.052	1.01	3.107	−0.79	3.057	0.83	3.049	1.08
Feb.	2.607	2.476	5.04	2.531	2.93	2.554	2.06	2.536	2.76	2.590	0.66	2,548	2.27	2.524	3.18
Mar.	2.237	2.117	5.38	2.152	3.81	2.202	1.57	2.174	2.84	2.232	0.23	2.195	1.89	2.164	3.29
Apr.	1.769	1.329	24.86	1.399	20.89	1.544	12.72	1.406	20.49	1.540	12.93	1.401	20.79	1.577	10.82
May	1.799	1.611	10.44	1.708	5.07	1.779	1.08	1.696	5.70	1.830	−1.73	1.715	4.67	1.716	4.64
Jun.	2.480	2.106	15.09	2.163	12.80	2.248	9.37	2.187	11.84	2.258	8.95	2.172	12.42	2.202	11.22
Aug.	3.803	3.632	4.48	3.676	3.34	3.712	2.39	3.695	2.84	3.734	1.81	3.692	2.91	3.675	3.36
Sep.	3.598	3.518	2.23	3.534	1.77	3.520	2.17	3.523	2.10	3.560	1.05	3.534	1.77	3.527	1.96
Oct.	3.107	2.983	3.99	3.043	2.07	3.047	1.93	3.055	1.68	3.114	−0.22	3.052	1.79	3.054	1.72
Nov.	3.449	3.398	1.47	3.412	1.06	3.428	0.60	3.422	0.78	3.450	−0.04	3.416	0.94	3.422	0.78

induced correction is marked with the letter I. The results are shown in Table I, where the unit of COST is 10^4.

The baseline obtained from SOCP indicates a significant difference in diverse scenarios. The solution from MISOCP, which is the absolute optimization space in each scenario, also varies greatly, verifying the rationality of the assumptions of PB. The test results show that the agent training based on the MBPO-B achieved the highest improvement, demonstrating better generalization capability. Additionally, the time for SOCP and MISOCP solving were 442.95 $\pm$ 78.34 s and 2232.43 $\pm$ 775.21 s, while the RL-based calculation time was 242.79 $\pm$ 3.45 s. Therefore, in long-term decision problems, the linearly increasing complexity of RL agents can achieve excellent suboptimal solutions in a shorter time.

The difference between the IMP between the RL policy test result and the corresponding MISOCP solution was plotted as shown in Fig. 7. Firstly, the generalizability of policy with EB modified is generally superior to that of uncorrected, regardless of the utilized RL algorithm. The difference in EB between

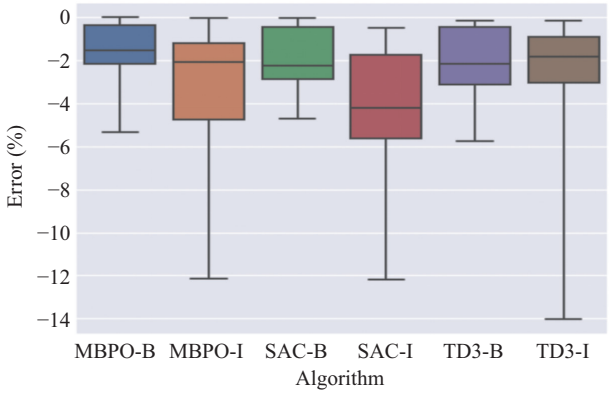

Fig. 7. Comparison of optimization capabilities for different algorithms.

different scenarios greatly impacts the estimation of the value function. Removing the influence of irrelevant policy rewards is essentially to refine the common features between different scenarios, which is conducive to improving generalization.

Meanwhile, the proposed algorithm performs the best among all the test cases, with the smallest average gap relative to the optimal and the smallest variance in gap distribution, proving its generalization and transfer capabilities. This is attributable to the sample enhancement effect of the learned dynamic model, which exposes the strategy to more potential state transitions during the training process, making it easier to devise reasonable strategies when encountering new scenarios.

F. Application of Model Rollouts

Figure 8 illustrates the training results as k increases with the training steps where k ranges from 1 to 5, with the update threshold at 60000 steps. The results indicate that when k lies between 1 and 3, the algorithm training successfully maintains improvement towards optimality. Remarkably, when k equals to 1, returns significantly higher than the average. However, when k exceeds 3, the policy exhibits substantial fluctuations, and policy updates disrupt the initially high returns. By monitoring the actual cost values of the system, it is evident that the system's cost rewards initially stabilize and subsequently exhibit a downward trend as training progresses. Consequently, the optimal rollout step should not exceed 3 when employing the learned model for sample augmentation.

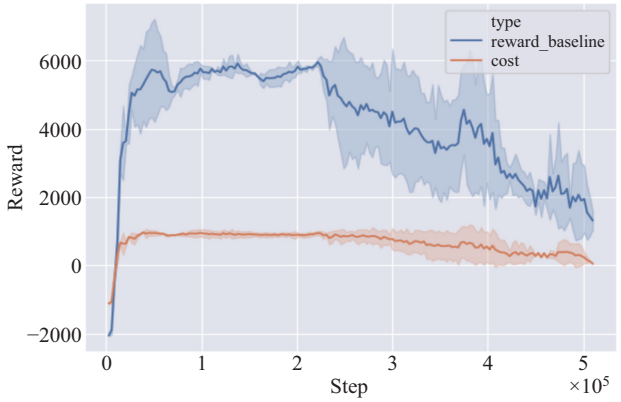

Fig. 8. Returns under different rollout step k.

VI. Conclusion

This paper proposes a DN optimization method based on the modified MBPO algorithm, which effectively enhances the efficiency of training and improves the generalization capability compared to model-free RL algorithms. The utilization of dynamic models essentially relies on the assumption of transition stability. However, practical scenarios present a multitude of uncontrollable external variables, including but not limited to RE and load factors. These factors pose a challenge as they cannot be accurately represented through a static transition model. This limitation is a key reason behind the sub-optimal performance when applying policy models to periods outside testing months.

In the future, it is imperative to conduct more in-depth studies into RL's transfer and generalization capabilities to enhance the stability of the strategies they generate. Moreover, It is also necessary to acknowledge that many practical requirements are difficult to articulate precisely in mathematical terms. Sole reliance on standard models is unlikely to yield better application. Therefore, it is essential to leverage the value of historical operational data from the grids, employing various learning methods to collaboratively support the realization of intelligent decision-making processes.

Appendix

A. Distribution Network Optimization Constraints

For DGs, in addition to rated and active power boundaries, the ramping constraints are also considered.

$$\begin{cases} P_i^{\mathrm{DG}} \le P_i^{\mathrm{DG}}(t) \le \bar{P}_i^{\mathrm{DG}} \\ -\eta_i^{\mathrm{DG}} \bar{P}_i^{\mathrm{DG}} \le P_i^{\mathrm{DG}}(t) - P_i^{\mathrm{DG}}(t-\Delta t) \le \eta_i^{\mathrm{DG}} \bar{P}_i^{\mathrm{DG}} \\ (P_i^{\mathrm{DG}}(t))^2 + (Q_i^{\mathrm{DG}}(t))^2 \le (\bar{S}_i^{\mathrm{DG}})^2 \end{cases} \tag{A1}$$

where $P^{\mathrm{DG}}, \bar{P}^{\mathrm{DG}}$ represents bounds of active power, η^{DG} denotes the maximum active change ratio, and $\bar{S}^{\mathrm{DG}}$ corresponds to the rated power. For RE, a constant power factor model is used.

$$\begin{cases} P_n^{\mathrm{RE}}(t) = \bar{P}_n^{\mathrm{RE}}(t) - P_n^{\mathrm{Cur}}(t) \\ Q_n^{\mathrm{RE}}(t) = \eta_n^{\mathrm{RE}} \bar{P}_n^{\mathrm{RE}}(t) \end{cases} \tag{A2}$$

where $\bar{P}^{\mathrm{RE}}(t)$ represents the limit of theoretical output of RE, and η^{RE} represents the constant power factor. The transmission primarily is used for purchasing from external grids.

$$\begin{cases} 0 \le P^{\mathrm{Ext}}(t) \le \bar{P}^{\mathrm{Ext}} \\ Q^{\mathrm{Ext}} \le Q^{\mathrm{Ext}}(t) \le \bar{Q}^{\mathrm{Ext}} \end{cases}, \forall t \in \Gamma \tag{A3}$$

where $\bar{P}^{\mathrm{Ext}}$ denotes the limit of purchased electricity, and $\bar{Q}^{\mathrm{Ext}}, Q^{\mathrm{Ext}}$ represents the bounds of reactive power interaction. For storage, auxiliary integer variables need to be constructed to limit the charging and discharging behavior.

$$\begin{cases} 0 \le P_j^{S,\mathrm{ch}}(t) \le I_j^{S,\mathrm{ch}}(t) \bar{P}_j^{S,\mathrm{ch}} \\ 0 \le P_j^{S,\mathrm{dis}}(t) \le I_j^{S,\mathrm{dis}}(t) \bar{P}_j^{S,\mathrm{dis}} \\ P_j^{S}(t) = I_j^{S,\mathrm{dis}}(t) P_j^{S,\mathrm{dis}}(t) - I_j^{S,\mathrm{ch}}(t) P_j^{S,\mathrm{ch}}(t) \\ I_j^{S,\mathrm{dis}}(t) + I_j^{S,\mathrm{ch}}(t) \le 1 \end{cases} \tag{A4}$$

$$\begin{cases} \mathrm{SoC}_j(t+\Delta t) = \mathrm{SoC}_j(t) + \eta_j^{S,\mathrm{ch}} P_j^{S,\mathrm{ch}}(t)/E_j^S \Delta t \\ -P_j^{S,\mathrm{dis}}(t)/\eta_j^{S,\mathrm{dis}} E_j^S \Delta t \\ \mathrm{SoC}_j^{\min} \le \mathrm{SoC}_j(t) \le \mathrm{SoC}_j^{\max} \end{cases} \tag{A5}$$

where $\bar{P}^{S,\mathrm{ch}}, \bar{P}^{S,\mathrm{dis}}$ are the limits of power, $I^{S,\mathrm{ch}}, I^{S,\mathrm{dis}}$ are the auxiliary binary integer variable, $\eta^{S,\mathrm{ch}}, \eta^{S,\mathrm{dis}}$ are the efficiency, E^S is the capacity.

$$\begin{cases} 0 \le P_k^{\mathrm{DR}}(t) \le \bar{P}_k^{\mathrm{DR}}(t) = \eta_k^{\mathrm{DR}} P_k^L(t) \\ Q_k^{\mathrm{DR}}(t) = Q_k^L(t) P_k^{\mathrm{DR}}(t)/P_k^L(t) \end{cases} \tag{A6}$$

The amount of load that can participate in DR occupies a fixed proportion. $\bar{P}^{\mathrm{DR}}$ is the limit of DR, η^{DR} is the proportion constant, and $P^L(t), Q^L(t)$ represents the load demand value.

The constraint of DN power flow is constructed with a branch flow model with balance relaxed, as shown in the following.

$$\begin{cases} P_j(t) = P_{ij}(t) - r_{ij} l_{ij}(t) - \sum_{k:(j,k)\in G^E} P_{jk}(t) \\ Q_j(t) = Q_{ij}(t) - x_{ij} l_{ij}(t) - \sum_{k:(j,k)\in G^E} Q_{jk}(t) \end{cases} \tag{A7}$$

$$v_j(t) = v_i(t) - 2(r_{ij}P_{ij}(t) + x_{ij}Q_{ij}(t)) + (r_{ij}^2 + x_{ij}^2) l_{ij}(t) \tag{A8}$$

$$V_i^{\min} \le |V_i(t)| \le V_i^{\max} \tag{A9}$$

where $l_{ij}(t)$ represents the square of branch current modulus, $v_i(t)$ represents the square of node voltage modulus, $V_i^{\max}, V_i^{\min}$ are the limits of the node voltage. (A10) represents the second-order cone convex relaxation constraints.

$$\left\| \begin{matrix} 2P_{ij}(t) \\ 2Q_{ij}(t) \\ l_{ij}(t) - v_i(t) \end{matrix} \right\|_2 \le l_{ij}(t) + v_i(t) \tag{A10}$$

B. Proof for Return Bounds

The TV distance and KL divergence can be linked through Pinsker's inequality; thus the TV divergence can also be bound. To prove Theorem1, need to prove lemma1 first.

Lemma 1: If the dynamics transition and policy distribution of two decision processes satisfies (22) and (23), then the cumulative return of the two processes satisfies (B1).

$$|\eta_1 - \eta_2| \le 2\gamma R \frac{\varepsilon_m + \varepsilon_\pi}{(1-\gamma)^2} \tag{B1}$$

To prove lemma1, it is necessary to use two fundamental conclusions proposed in reference (20), the contents of which are introduced here without detailed proof.

Lemma 2: Assuming two different joint probability distributions are $p_1(x,y), p_2(x,y)$, their TV distance satisfies the following inequality.

$$\begin{aligned} &D_{\mathrm{TV}}(p_1(x,y)\|p_2(x,y)) \\ &\le D_{\mathrm{TV}}(p_1(x)\|p_2(x)) + \max_x D_{\mathrm{TV}}(p_1(y|x)\|p_2(y|x)) \end{aligned} \tag{B2}$$

Lemma 3: Assuming the state transitions are $p_1(s'|s)$, $p_2(s'|s)$ and their KL divergence constraint is δ, then its state probability satisfies the following inequality (B3).

$$D_{\mathrm{TV}}(p_1^t(s)\|p_2^t(s)) \le t\delta \tag{B3}$$

Lemma 1 Proof. The conclusion of Lemma 1 is proven based on Lemma 2 and Lemma 3. Firstly, the policy return is expanded to obtain formula A.13.

$$\begin{aligned} |\eta_1 - \eta_2| &= \Big| \sum_{s,a,s'} (p_1(s,a,s') - p_2(s,a,s')) r(s,a,s') \Big| \\ &= \Big| \sum_t \sum_{s,a,s'} \gamma^t (p_1(s,a,s') - p_2(s,a,s')) r(s,a,s') \Big| \\ &\le |r(s,a,s')|_{\max} \sum_t \sum_{s,a,s'} \gamma^t |p_1(s,a,s') - p_2(s,a,s')| \end{aligned} \tag{B4}$$

where $p(s,a,s')$ can be rewritten as the conditional probability form. the probability condition can be simplified.

$$\begin{aligned} \sum_{s,a,s'} p(s,a,s') &= \sum_{s,a,s'} p(s'_a|s_a,a) p(s'_{\mathrm{ext}}|s_{\mathrm{ext}}) p(s,a) \\ &= \sum_{s,a,s'} p(s'_{\mathrm{ext}}|s_{\mathrm{ext}}) p(s,a) = \sum_{s,a,s'} p(s'_{\mathrm{ext}}, s, a) \end{aligned} \tag{B5}$$

Thus, the distribution bias can be further rewritten, and Lemma 2 is applied twice to obtain its upper bound.

$$\begin{aligned} &\sum_{s,a,s'} |p_1(s,a,s') - p_2(s,a,s')| \\ &\le 2\max_{s,a} D_{\mathrm{TV}}(p_1(s'_{\mathrm{ext}}|s_{\mathrm{ext}}) \| p_2(s'_{\mathrm{ext}}|s_{\mathrm{ext}})) \\ &\quad + 2\max_{s,a} D_{\mathrm{TV}}(p_1(a|s) \| p_2(a|s)) + 2 D_{\mathrm{TV}}(p_1(s) \| p_2(s)) \end{aligned} \tag{B6}$$

The first two terms of formula (B6) correspond to the Lemma 2, and the third term needs to apply Lemma 3 where $\delta = \varepsilon_m + \varepsilon_\pi$. Therefore, the result can be further rewritten as follows:

$$\sum_t \sum_{s,a,s'} \gamma^t |p_1(s,a,s') - p_2(s,a,s')| \le \frac{2\gamma(\varepsilon_m + \varepsilon_\pi)}{(1-\gamma)^2} \tag{B7}$$

Multiplying the coefficient $r_{\max}$ by the result of formula (B7) can prove that Lemma 1 holds. Apparently, the r^{env} must be negative, the r^π must be positive, and the sum $r^\pi + r^{\mathrm{env}}$ must be negative. The inequality as shown in (B8) can be obtained.

$$|r_t^\pi(s_t,a_t) + r_t^{\mathrm{env}}(s_t^{\mathrm{ext}})| < |r_t^{\mathrm{env}}(s_t^{\mathrm{ext}})| \le \max_s |r^{\mathrm{env}}(s^{\mathrm{ext}})| \tag{B8}$$

Add the bound of r^{ext} after A.17, and finally get the upper bound as shown in Lemma 1. Refer to the proving method of reference [24], the deviation of the whole return bound is shown in A.18 based on Lemma 1. Multiply by $r_{\max}$ to finish the Theorem 1 proof.

$$\begin{aligned} &|\eta(\pi) - \hat{\eta}(\pi)| = |\eta(\pi) - \eta(\pi_D) + \eta(\pi_D) - \hat{\eta}(\pi)| \\ &\le |\eta(\pi) - \eta(\pi_D)| + |\eta(\pi_D) - \hat{\eta}(\pi)| \leftarrow \text{Lemma 1} \\ &\le \frac{2r_{\max}\gamma\varepsilon_\pi}{(1-\gamma)^2} + \frac{2r_{\max}\gamma(\varepsilon_m + \varepsilon_\pi)}{(1-\gamma)^2} = 2\gamma r_{\max} \frac{\varepsilon_m + 2\varepsilon_\pi}{(1-\gamma)^2} \end{aligned} \tag{B9}$$

C. Detailed Parameters of Controllable Factors

TABLE CI

Element	Capacity	Coefficient	Bus	Other
DG	0.8	α : 30 β : 25 c: 0.2	24	$\cos\theta = 0.8$
	0.8	α : 75 β : 18 c : 0.2	94	$\cos\theta = 0.8$
	0.8	α : 40 β : 22 c : 0.2	114	$\cos\theta = 0.8$
PV	0.4	ρ : 60	22, 123	Controllable
	0.2	–	39, 41, 121	Uncontrollable
WT	0.6	ρ : 80	59	Controllable
	0.2	–	4, 46, 75, 83	Uncontrollable

TABLE CI
(CONTINUED)

Element	Capacity	Coefficient	Bus	Other
DR	0.15	ζ : 20	66	–
	0.1	ζ : 24	90	–
	0.12	ζ : 23	107	–
BS	$\pm$ 0.3 $E^S = 4$	σ : 0.2	20	η^{ch} : 0.95, η^{dis} : 0.9 SoC $\in [10\%, 90\%]$
	$\pm$ 0.2 $E^S = 4$	σ : 0.4	56	

D. Normalized Data Profiles of Uncontrollable Factors

Figure D1 shows one of the environmental patterns in the study case. PV and WT data contain a large number of power fluctuation scenarios. Among them, the load, PV and WT data are normalized.

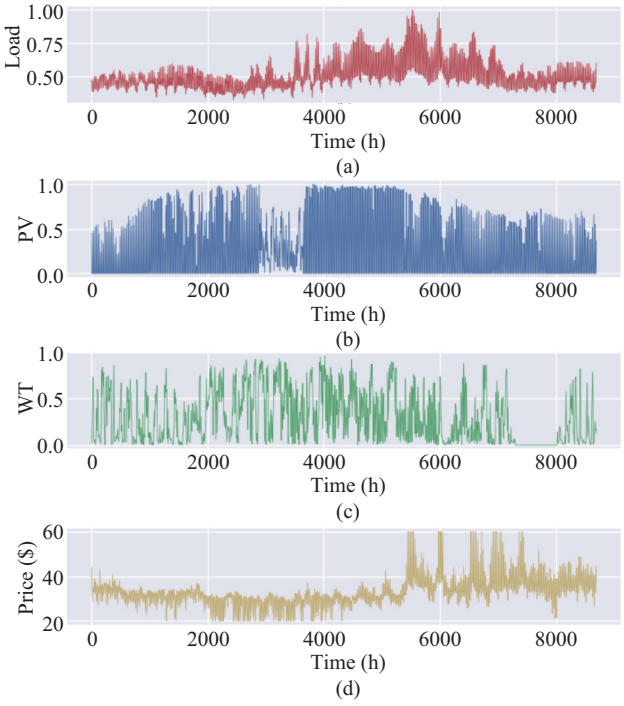

Fig. D1. The normalized data profiles of (a) load, (b) PV, (c) WT and (d) price in study case.

REFERENCES

[1] J. Y. Wang, "Application and prospect of source-grid-load-storage coordination enabled by artificial intelligence," *Proceedings of the CSEE*, vol. 42, no. 21, pp. 7667–7681, Nov. 2022.

[2] Z. N. Li, S. Su, X. L. Jin, M. C. Xia, Q. F. Chen, and K. Yamashita, "Stochastic and distributed optimal energy management of active distribution networks within integrated office buildings," *CSEE Journal of Power and Energy Systems*, vol. 10, no. 2, pp. 504–517, Mar. 2024.

[3] Q. Y. Sun, B. Y. Wang, X. M. Feng, and S. Y. Hu, "Small-signal stability and robustness analysis for microgrids under time-constrained DoS attacks and a mitigation adaptive secondary control method," *Science China Information Sciences*, vol. 65, no. 6, pp. 162202, Apr. 2022.

[4] R. Hu and Q. F. Li, "Optimal operation of power systems with energy storage under uncertainty: a scenario-based method with strategic sampling," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1249–1260, Mar. 2022.

[5] W. J. Li, Y. G. Liu, H. J. Liang, Y. C. Man, and F. Z. Li, "Distributed tracking-ADMM approach for chance-constrained energy management with stochastic wind power in smart grid," *CSEE Journal of Power and Energy Systems*, doi: 10.17775/CSEEJPES.2021.00540.

[6] Z. D. Zhang, D. X. Zhang, and R. C. Qiu, "Deep reinforcement learning for power system applications: an overview," *CSEE Journal of Power and Energy Systems*, vol. 6, no. 1, pp. 213–225, Mar. 2020.

[7] T. J. Wang, Y. Tang, Q. Guo, Y. H. Huang, X. L. Chen, and H. K. Huang, "Automatic adjustment method of power flow calculation convergence for large-scale power grid based on knowledge experience and deep reinforcement learning," *Proceedings of the CSEE*, vol. 40, no. 8, pp. 2396–2406, Apr. 2020.

[8] T. J. Wang and 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.

[9] 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.

[10] J. Yang, J. Liu, Y. Xiang, S. Zhang and J. Liu, "Data-driven Optimal Dynamic Dispatch for Hydro-PV-PHS Integrated Power Systems Using Deep Reinforcement Learning Approach," *CSEE Journal of Power and Energy Systems*, vol. 9, no. 3, pp. 846–858, May 2023.

[11] D. Wang, B. Liu, H. J. Jia, Z. Y. Zhang, J. C. Chen, and D. Y. Huang, "Peer-to-peer electricity transaction decisions of the user-side smart energy system based on the SARSA reinforcement learning," *CSEE Journal of Power and Energy Systems*, vol. 8, no. 3, pp. 826–837, May 2022.

[12] Y. Zhang, Q. Ai and Z. Li, "Intelligent Demand Response Resource Trading Using Deep Reinforcement Learning," *CSEE Journal of Power and Energy Systems*, vol. 10, no. 6, pp. 2621–2630, Nov. 2024

[13] J. Qiao, X. Y. Wang, Q. Zhang, D. X. Zhang, and T. J. Pu, "Optimal dispatch of integrated electricity-gas system with soft actor-critic deep reinforcement learning," *Proceedings of the CSEE*, vol. 41, no. 3, pp. 819–832, Feb. 2021.

[14] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 1861–1870.

[15] L. Y. Peng, Y. Z. Sun, J. Xu, S. Y. Liao, and L. Yang, "Self-adaptive uncertainty economic dispatch based on deep reinforcement learning," *Automation of Electric Power Systems*, vol. 44, no. 9, pp. 33–42, May 2020.

[16] L. Yang, Q. Sun, N. Zhang, and Y. S. Li, "Indirect multi-energy transactions of energy internet with deep reinforcement learning approach," *IEEE Transactions on Power Systems*, vol. 37, no. 5, pp. 4067–4077, Sep. 2022.

[17] Y. Ji, J. H. Wang, J. C. Xu, X. K. Fang, and H. G. Zhang, "Real-time energy management of a microgrid using deep reinforcement learning," *Energies*, vol. 12, no. 12, pp. 2291, Jun. 2019.

[18] H. P. Li and H. B. He, "Learning to operate distribution networks with safe deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 13, no. 3, pp. 1860–1872, May 2022.

[19] Y. Ji and J. H. Wang, "Online optimal scheduling of a microgrid based on deep reinforcement learning," *Control and Decision*, vol. 37, no. 7, pp. 1675–1684, Jul. 2022.

[20] H. Shuai and H. B. He, "Online scheduling of a residential microgrid via Monte-Carlo tree search and a learned model," *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 1073–1087, Mar. 2021.

[21] S. H. Gao, C. Xiang, M. Yu, K. T. Tan, and T. H. Lee, "Online optimal power scheduling of a microgrid via imitation learning," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 861–876, Mar. 2022.

[22] R. K. Huang, Y. J. Chen, T. Z. X. Yin, Q. H. Huang, J. Tan, W. H. Yu, X. Y. Li, A. Li, and Y. Du, "Learning and fast adaptation for grid emergency control via deep meta reinforcement learning," *IEEE Transactions on Power Systems*, vol. 37, no. 6, pp. 4168–4178, Nov. 2022.

[23] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: model-based policy optimization," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2019.

[24] S. H. Low, "Convex relaxation of optimal power flow—Part I: formulations and equivalence," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 15–27, Mar. 2014.

[25] F. M. Luo, T. Xu, H. Lai, X. H. Chen, W. N. Zhang, and Y. Yu, "A survey on model-based reinforcement learning," arXiv: 2206.09328, 2022.

[26] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: learning behaviors by latent imagination," in *Proceedings of the International Conference on Learning Representations*, 2020.

[27] OASIS. (2021). California ISO open access same-time information system. [Online]. Available: http://oasis.caiso.com/mrioasis/logon.do

Dong Yan received the M.Sc. degree in Electrical and Computer Engineering in 2018, from University of Michigan, Ann Arbor, USA. He is currently working towards a Ph.D. degree in Power Systems and Automation with China Electric Power Research Institute, Beijing, China. His research interests encompass the application of artificial intelligence in power systems, focusing on the utilization of deep reinforcement learning and its derivative methodologies for the optimization of power systems.

Zhan Shi received the Ph.D. degree in Control Theory and Control Engineering from Northeastern University, Shenyang, China, in 2022. His current research interests include adaptive control, optimal control, economic dispatch and their applications in power system, smart grid and microgrids.

Xinying Wang received his Ph.D. degree at Dalian University of Technology, Dalian, China, in 2015 and has worked in China Electric Power Research Institute, Beijing, China, until now. He is the Director of Big Data Application Research Section of CEPRI, the Senior Member of CSEE. His main fields of interest are artificial intelligence and its application in energy internet.

Yiying Gao received the M.S. degree in Electrical Engineering from Harbin Institute of Technology, Harbin, China in 2018. She currently works in China Electric Power Research Institute. Her research interest is power system optimization.

Tianjiao Pu received the M.S. degree in Electrical Engineering from Tianjin University in 1997. He is the Director of Artificial Intelligence Application Research Department of China Electric Power Research Institute, Fellow of IET, Senior Member of IEEE, and Senior Member of CSEE. He has been engaged in the research and management of power dispatching automation, smart grid simulation, active distribution network, artificial intelligence and other fields.

Jiye Wang received the M.S. degree in Electrical Engineering and Automation from the North China Electric Power University, Beijing. He is currently the Director of State Grid Digital Technology Holding Co., Ltd., the Ph.D. Supervisor, and the Fellow of CSEE. His research interests include energy Internet and smart grid.