Dynamic Weighted Multi-Agent Reinforcement Learning with Hybrid Reward Mechanism for Microgrid Energy Scheduling

Ruijie Ma, Jianhua Li*, Dekai Huang, Kezhi Lu, Zhaoming Qin, Weijian Ou, Junnan Wu
Power Workshop, Nanning Cigarette Factory, Nanning Guangxi, 530000, China
E-mail: 17607738386@163.com
*Corresponding author

Keywords: multi-agent reinforcement learning; microgrid, energy scheduling; dynamic weight allocation; reward function

Received: August 5, 2025

With growing energy demand and increasing environmental concerns, optimizing energy scheduling is crucial for microgrids, as efficient and flexible energy systems. This paper proposes an innovative multi-agent reinforcement learning algorithm to address microgrid energy scheduling. This algorithm incorporates a dynamic weight allocation mechanism, enabling agents to flexibly adjust decision weights based on real-time changes in energy supply and demand, improving the adaptability of scheduling strategies. Furthermore, a reward function based on historical data and real-time status is designed to guide agents in learning optimal energy scheduling strategies. To validate the algorithm's effectiveness, a microgrid energy scheduling simulation platform was constructed to simulate energy production, consumption, and storage in various scenarios. Experimental results show that, compared with traditional algorithms, the proposed algorithm can improve microgrid energy efficiency by an average of 15% and reduce operating costs by 12%. In scenarios with a high proportion of renewable energy, the algorithm effectively reduces energy waste and increases renewable energy consumption by 20%. Furthermore, stability analysis demonstrates that the algorithm maintains stable scheduling performance in the face of energy supply and demand uncertainties, showing strong robustness. This study provides a new solution for microgrid energy scheduling, helping to improve its overall performance and sustainability.

Povzetek: Članek predlaga večagentsko učenje z ojačitvami za razporejanje energije v mikroomrežju z dinamično dodelitvijo uteži in nagrajevanjem, ki združuje zgodovinske in sprotne podatke, s čimer prilagodi odločitve, izboljša izrabo obnovljivih virov ter robustnost razporejanja.

1 Introduction

As small-scale energy networks integrating distributed energy resources, energy storage systems, and diverse loads, microgrids play a crucial role in supporting the transformation of energy structures. Driven by the "dual carbon" goals, the global installed capacity of microgrids has grown at an average annual rate of over 12%, with the proportion of renewable energy increasing year by year, exceeding 50% in some regions. However, the integration of high-proportion renewable energy renders microgrids highly random and volatile. Traditional model-based scheduling methods, which rely on precise mathematical models, are incapable of addressing supply-demand imbalances caused by unexpected events such as cloud cover and sudden changes in wind speed. This, in turn, leads to issues including reduced energy utilization efficiency and increased operational costs [1]. Actual operational data indicates that under existing scheduling strategies, the curtailment rate of renewable energy often exceeds 15% when the penetration rate of wind and solar power surpasses 30%. Fluctuations in peak and valley loads result in redundancy rates of over 20% for equipment such as transformers and transmission lines, leading to annual economic losses accounting for more than 8% of the total system investment [2]. Multi-Agent Reinforcement Learning (MARL) offers a new paradigm for scheduling complex energy systems through distributed decision-making and collaborative optimization. Its core advantage lies in achieving dynamic optimization through trial-and-error learning without the need for precise models [3], and it has demonstrated application potential in areas such as power system voltage control and demand response. Nevertheless, existing research still has significant limitations: most schemes adopt static coordination mechanisms with fixed agent decision weights, making it difficult to adapt to sudden changes in microgrid operating conditions [4]. For example, the MADDPG algorithm, which is based on a centralized critic, experiences scheduling errors exceeding 30% when load fluctuations exceed 25% due to the overload of global state information. The IQL algorithm, utilizing an independent learning framework, suffers from agent strategy convergence conflicts in energy-complementary scenarios due to the lack of a coordination mechanism, resulting in a convergence speed reduction of more than 40% [5]. Additionally, the uniqueness of microgrid scheduling imposes higher requirements on MARL algorithms: on one hand, the dynamic characteristics of source-storage-load devices vary significantly— the response time of photovoltaic output is measured in minutes, and the charging and discharging efficiency of energy storage systems changes nonlinearly with the State of Charge (SOC), necessitating differentiated decision-making mechanisms; on the other hand, the temporal and spatial distribution characteristics of commercial, industrial, and residential loads differ, with the overlap between peak and valley periods being less than 30%, which increases the difficulty of collaborative optimization. Most existing algorithms employ a unified reward function and decision-making model, making it challenging to balance device characteristics and load differences, leading to a deviation of more than 15% between the actual scheduling effect and the theoretical optimal solution [6]. To address the aforementioned challenges, this study proposes a MARL scheduling algorithm that integrates dynamic weight allocation and a hybrid reward mechanism. The specific research questions are defined as follows:

Question 1: Can the dynamic weight allocation mechanism enhance the algorithm's adaptability to renewable energy fluctuations and reduce scheduling errors caused by supply-demand imbalances?

Question 2: Can a hybrid reward function that integrates historical data trends and real-time status solve the problem of low learning efficiency under sparse rewards and promote agents to learn optimal scheduling strategies?

Question 3: Can the proposed algorithm maintain stable energy efficiency, cost control, and renewable energy absorption performance in large-scale microgrid scenarios with multiple disturbances?

The research objectives include: (1) Constructing a distributed scheduling framework based on multi-agent collaboration to achieve the collaborative optimization of source-storage-load systems; (2) Designing an adaptive weight adjustment strategy to improve the algorithm's robustness against renewable energy fluctuations; (3) Verifying the performance advantages of the algorithm in multiple scenarios through simulation experiments. The innovations are reflected in: 1) Proposing a dynamic weight allocation mechanism based on real-time operating conditions, enabling the decision weights of agents to be dynamically adjusted according to the energy supply-demand ratio, thereby addressing the insufficient adaptability of static weights; 2) Constructing a hybrid reward function that integrates historical data trends and real-time status, and solving the learning efficiency issue under sparse rewards by introducing long-term and short-term benefit factors; 3) Establishing a microgrid simulation scenario with uncertain disturbances to quantify the stability gains of the algorithm under extreme operating conditions, providing data support for practical engineering applications.

2 Related theoretical foundations

The integration of microgrid energy scheduling and multi-agent reinforcement learning requires a solid theoretical framework [7]. This chapter constructs a theoretical framework based on the characteristics of microgrid systems and the collaborative mechanisms of intelligent agents. Original formulas are used to quantify core relationships, providing a mathematical basis for algorithm design. The practical significance of the theory is also illustrated through practical application scenarios.

2.1 Fundamentals of microgrid energy scheduling

As a typical form of distributed energy system, the scheduling process of a microgrid must balance the dynamic relationship between multi-energy complementarity and system constraints, involving the coordinated optimization of energy conversion, storage, and distribution [8]. This optimization must not only consider the real-time balance between energy supply and demand, but also the system's economic efficiency, stability, and environmental performance. It is a complex multi-objective optimization problem.

2.1.1 Microgrid structure and composition

A microgrid consists of distributed generation units (DG), energy storage systems (ESS), controllable loads (CL), and energy conversion equipment. DG includes intermittent power sources such as photovoltaics (PV) and wind turbines (WT), and controllable power sources such as micro turbines (MT). The output of intermittent power sources is significantly affected by natural conditions and has high uncertainty, while controllable power sources can flexibly adjust their production according to scheduling needs. The ESS utilizes a hybrid lithium battery and flywheel energy storage architecture. Lithium batteries have high energy density and are suitable for long-term energy storage. In contrast, flywheel energy storage has high power density and fast response speed, enabling hierarchical management of power and energy to meet different energy storage requirements [9]. CL is divided into two categories: shiftable loads (such as electric vehicle charging) and curtailable loads (such as air conditioning temperature control). Shiftable loads can adjust power consumption within a specific period, while curtailable loads can appropriately reduce power consumption when necessary. Each unit is connected to the DC bus via a power electronic converter, forming a closed-loop "source-storage-load" system. Its topology supports both island and grid-connected operation. In grid-connected mode, it can exchange energy with the primary grid, while in island mode, it relies on its energy supply to maintain stable operation.

2.1.2 Key elements and constraints of energy scheduling

The core of scheduling optimization is to maximize energy efficiency while satisfying system constraints. The power balance constraint at time t is defined as:

$$\sum_{i=1}^{n} P_{DG,i}(t) + P_{ESS,dch}(t) - P_{ESS,ch}(t) = \sum_{j=1}^{m} P_{L,j}(t) + P_{loss}(t) \tag{1}$$

Where $P_{DG,i}(t)$ is the output of the i-th type of distributed generation, $P_{ESS,ch/dch}(t)$ is the energy storage charge/discharge power, $P_{L,j}(t)$ is the load power of the j type, and $P_{\text{loss}}(t)$ is the line loss. This constraint is the basic premise for the stable operation of the microgrid and must be met at all times [10]. Otherwise, it may cause voltage and frequency fluctuations and affect the normal operation of the power-consuming equipment. The energy storage system must meet the SOC constraint:

$$SOC_{\min} \le SOC(t) + \eta_{ch} P_{ESS,ch}(t)\Delta t - \frac{P_{ESS,dch}(t)\Delta t}{\eta_{dch}} \le SOC_{\max} \tag{2}$$

Where η_{ch}/η_{dch} is the charge/discharge efficiency and Δt is the scheduling step size. The SOC constraint ensures that the energy storage system operates within a safe range, avoids damage to the energy storage equipment caused by overcharging or overdischarging, and prolongs its service life. To quantify the complementarity of multiple energy sources, the flexibility coefficient is defined:

$$\gamma(t) = \frac{\sum_{i=1}^{n} |P_{DG,i}(t) - \bar{P}_{DG,i}|}{\sum_{j=1}^{m} |P_{L,j}(t) - \bar{P}_{L,j}|} \tag{3}$$

Where $\bar{P}_{DG,i}\ and\ \bar{P}_{L,j}$ are historical average outputs. When $\gamma(t) < 1$, power fluctuations are smaller than load fluctuations, requiring energy storage buffering. When $\gamma(t) > 1$, power fluctuations are larger, requiring load adjustment or interaction with the larger grid to maintain balance. The flexibility coefficient provides an important reference for scheduling strategy formulation, helping to assess system stability and regulation difficulty.

2.2 Principles of multi-agent reinforcement learning

Multi-agent systems achieve complex task collaboration through distributed decision-making. Their reinforcement learning process must address policy optimization and credit allocation in a dynamic environment [11]. In microgrid energy scheduling, multiple agents correspond to different energy units and must collaborate to achieve overall optimization goals.

2.2.1 Concept and characteristics of agents

Agents are autonomous entities with perception, decision-making, and execution capabilities. In a microgrid, they serve as control units for sources, storage, and loads. Its core characteristics are: state perception capability $S_a = f_o(O_a)$ (O_a is the observed value), enabling real-time information on its own and surrounding states; action output capability $A_a = \pi_a(S_a \mid \theta_a)$ (θ_a is the policy parameter), enabling decision-making based on the perceived state; and interactive communication capability $M_a = f_c(S_a, M_a)$ (M_a is messages from other agents), enabling information exchange and collaboration with other agents.

For energy scheduling scenarios, agents must possess spatiotemporal correlation perception, namely:

$$S_a(t) = \alpha S_a(t-1) + (1-\alpha) f_s\big(O_a(t), O_a(t)\big) \tag{4}$$

Here, α is the historical state decay factor, which determines the degree of influence of historical states on the current state. $f_s(\cdot)$ is the spatial correlation function, reflecting the agent's perception of the states of neighboring cells. This spatial and temporal correlation allows the agent to comprehensively consider historical experience and the influence of the surrounding environment, making more rational decisions.

2.2.2 Basic framework of reinforcement learning

Reinforcement learning models the interaction between the agent and the environment using a Markov decision process (MDP). Define the state space S, the action space A, the reward function R, and the state transition probability P. The agent's policy $\pi(a \mid s)$ aims to maximize the cumulative reward:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{5}$$

Where $\gamma \in [0,1)$ is the discount factor. The magnitude of the discount factor reflects the importance placed on future rewards; larger values indicate greater emphasis on future long-term rewards.

The value function $V^{\pi}(s) = \mathbb{E}[G_t \mid S_t = s]$ measures the value of the state, and the advantage function $A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$ evaluates the relative value of actions. In the continuous action space, a Gaussian policy $\pi(a \mid s) = \mathcal{N}(\mu_\theta(s), \sigma_\theta^2(s))$ is used, and the parameters are updated using the policy gradient method:

$$\nabla_\theta J(\theta) = \mathbb{E}[\nabla_\theta \log \pi(\, a \mid s\,)\, A^{\pi}(s,a)] \tag{6}$$

The policy gradient method continuously adjusts policy parameters to enable agents to learn the optimal decision-making strategy to maximize cumulative rewards gradually.

2.2.3 Characteristics of reinforcement learning in multi-agent environments

Multi-agent reinforcement learning (MARL) faces the challenge of non-stationary environments. Changes in the policies of other agents cause the probability of transitions, P, to vary over time. This non-stationarity increases the difficulty of learning, requiring agents to adapt to changes in the environment constantly. Define joint states $S^j = \{S_1, \ldots, S_N\}$ and joint actions $A^j = \{A_1, \ldots, A_N\}$. The local reward R_i of agent i is mapped to the global reward R^g by $R_i = \beta_i R^g + (1 - \beta_i) r_i^l$ (where β_i is the global reward weight and r_i^l is the local reward).The credit allocation problem, which involves decomposing each agent's contribution from R^g, is a key issue in multi-agent collaboration. To solve this problem, it is necessary to design a value decomposition network based on the attention mechanism to achieve $Q^j(S^j, A^j) = \sum_{i=1}^{N} \omega_i(S^j) Q_i(S_i, A_i)$, where $\omega_i(S^j)$ is a dynamic weight coefficient, which can dynamically adjust the value weight of each intelligent agent according to the joint state, making credit distribution more reasonable and accurate.

2.3 Comparative analysis of related research

To further clarify the differences between the algorithm proposed in this study and existing research, Table 1 compares mainstream MARL microgrid scheduling algorithms from five dimensions: core technology, reward mechanism, weight strategy, applicable scenarios, and limitations, highlighting the innovation and necessity of the proposed scheme.

Table 1: Comparison of mainstream MARL microgrid scheduling algorithms

Algorithm Name	Core Technology	Reward Mechanism	Weight Strategy	Applicable Scenarios	Limitations
MADDPG [4]	Centralized critic network + distributed execution	Single-dimensional reward based on real-time power balance	No explicit weight allocation, relying on global state fusion	Small-scale microgrids with low fluctuations	Overload of global state information, resulting in scheduling errors exceeding 30% when load fluctuations exceed 25%
IQL [5]	Independent learning framework + local value network	Local reward considering only the operating cost of local equipment	Equal agent weights without collaborative adjustment	Scenarios with single energy type and stable loads	Lack of a coordination mechanism, leading to a convergence speed reduction of more than 40% in energy-complementary scenarios
DDPG [16]	Single-agent Actor-Critic architecture	Dual-objective reward combining energy efficiency and cost	No multi-agent weight concept, with decisions dominated by a single agent	Medium-scale scenarios with renewable energy proportion < 20%	Inability to balance the characteristic differences of multiple devices, resulting in a renewable energy curtailment rate exceeding 18% in high wind-solar penetration scenarios
SW-MARL [18]	Multi-agent collaboration + static weight allocation	Short-term reward based on real-time load demand	Initial weights determined by equipment capacity ratio, fixed during operation	Scenarios with load fluctuations < 15% and few equipment failures	Inability to dynamically adjust weights, leading to a 25% performance degradation when wind-solar fluctuations exceed 20%

Proposed DW-MARL	Dynamic weight allocation + hybrid reward mechanism + federated learning collaboration	Multi-dimensional reward integrating historical trends, real-time status, and equipment aging			

3 Multi-agent reinforcement learning algorithm design

This chapter focuses on the design of a multi-agent reinforcement learning algorithm suitable for microgrid scheduling [12]. By building a dynamic collaboration framework and adaptive learning mechanism, it achieves coordinated optimization of the source-storage-load system. The algorithm design balances distributed decision-making efficiency with global optimization goals. Core innovations lie in the dynamic adaptation of agent roles, adaptive weight adjustment, and a hybrid reward mechanism.

3.1 Overall algorithm architecture

The algorithm adopts a two-layer architecture of "distributed decision-making and centralized evaluation." The bottom layer consists of individual energy unit agents, and the upper layer houses a coordinator for global state fusion and policy evaluation.

3.1.1 Agent roles and division of labor

Agents are classified into four categories based on the type of energy unit. The input-output features and action spaces of each agent are clearly defined as follows:

- **Photovoltaic agent (PVA)**: Inputs include light intensity (400-1000 W/m²), ambient temperature (-5-45°C), and historical output data (previous 12 scheduling steps); outputs consist of the photovoltaic maximum power point tracking coefficient and the output prediction correction value; the action space is continuous, including 2 action dimensions (tracking coefficient adjustment step: -0.05-0.05, prediction correction coefficient: 0.8-1.2), responsible for maximum power point tracking and output prediction correction.

- **Wind turbine agent (WTA)**: Inputs include real-time wind speed (0-25 m/s), wind direction angle (0-360°), and wind turbine speed (500-1500 r/min); output is the nonlinear compensation coefficient for wind speed-power conversion; the action space is continuous, containing 1 action dimension (compensation coefficient: 0.9-1.1), responsible for handling the nonlinear compensation of wind speed-power conversion.

- **Energy storage agent (ESA)**: Inputs include current SOC (20%-80%), health status (SOH: 80%-100%), and charging-discharging efficiency (0.85-0.95); outputs are the charging-discharging power command (-100-100 kW) and charging-discharging duration; the action space is continuous, with 2 action dimensions (power command: -100-100 kW, duration: 15-60 min), responsible for managing the charging-discharging depth and balancing lifespan loss.

- **Load agent (LA)**: Inputs include load type (industrial/commercial/residential), real-time power demand (0-120 kW), and shiftable load time period (0-24 h); outputs are the demand response priority ranking and the curtail able load ratio; the action space is a discrete-continuous hybrid, including 2 action dimensions (priority: 1-5 levels, curtailment ratio: 0-0.3), responsible for executing demand response priority ranking.

Each agent obtains state variables through a local observer, and the agent capability coefficient is defined as: $\lambda_a(t) = \lambda_1 \cdot \frac{P_{a+ail,a}(t)}{P_{\max,a}} + \lambda_2 \cdot \frac{1\gamma(t)\gamma_a(t)}{1}$. where $\lambda_1 + \lambda_2 = 1(\lambda_1 = 0.6, \lambda_2 = 0.4)$, which quantifies the task suitability of the agent under current operating conditions; $P_{\max,a}$ is the maximum output/load capacity of the equipment corresponding to the agent, and $\gamma_a(t)$ is the flexibility coefficient of the equipment corresponding to the agent.

3.1.2 Overview of multi-agent collaboration mechanism

- Agents share parameters through a federated learning framework and adopt an "event-triggered, on-demand communication" model to reduce interaction overhead. When the fluctuation of the system flexibility coefficient

$\gamma(t)$ exceeds the threshold of $\pm 15\%$, the collaborative decision-making process is triggered:

$$\text{Trigger} = \begin{cases} 1, & |\gamma(t) - \gamma(t-1)| \geq 0.15 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Details of the Federated Learning Mechanism: The Federated Averaging (FedAvg) algorithm is used to implement model aggregation, with the specific process as follows: 1) In the local training phase, each agent updates the model parameters based on local data, with the number of training rounds set to 5; 2) In the model upload phase, agents encrypt and upload local model parameters to the coordinator only when collaborative decision-making is triggered (Trigger=1) or the number of local training iterations reaches 10; 3) In the global aggregation phase, the coordinator calculates the weighted average parameters based on the dynamic weight $\omega_a(t)$ of each agent to update the global model; 4) In the model distribution phase, the updated global model parameters are distributed to each agent to complete synchronization. This mechanism can reduce ineffective data transmission by more than 90% and lower communication overhead.

During the collaboration process, the coordinator aggregates local decisions through an attention mechanism:

$$a_g = \sum_{a=1}^{N} \alpha_a \cdot a_a, \alpha_a = \frac{\exp\left(\text{Attn}(s_a, s_g)\right)}{\sum_{b=1}^{N} \exp\left(\text{Attn}(s_b, s_g)\right)} \tag{8}$$

where $\text{Attn}(\cdot)$ is the attention scoring function (using an additive attention mechanism: $\text{Attn}(s_a, s_g) = W_v^T \tanh\left(W_k s_a + W_q s_g\right)$, with W_k, W_q, W_v being learnable parameter matrices), and s_g is the global state vector (integrating the state of each agent, system power balance deviation, and the proportion of renewable energy output).

3.2 Dynamic weight allocation mechanism

3.2.1 Weight allocation basis and method

The initial weight is calculated using an improved entropy weight method, taking into account device capacity ratios and historical scheduling contributions:

$$\omega_a^{(0)} = \frac{1 - H_a}{\sum_{b=1}^{N} (1 - H_b)}, H_a = -\sum_{t=1}^{T} \frac{p_{a,t}}{\sum_{t=1}^{T} p_{a,t}} \ln\left(\frac{p_{a,t}}{\sum_{t=1}^{T} p_{a,t}}\right) \tag{9}$$

where H_a is the historical decision entropy of agent a (a lower entropy value indicates higher stability and effectiveness of historical decisions), and $p_{a,t}$ is the decision influence of agent a at time t (defined as the correction ratio of the agent's decision to the system power balance deviation at that time).

3.2.2 Dynamic weight adjustment strategy

Real-time weight adjustment introduces a supply-demand ratio feedback term and a capacity decay factor:

$$\begin{aligned} \omega_a(t) &= \omega_a(t-1) \cdot \left[1 + \delta \cdot \left(\frac{\Delta P_a(t)}{\sum_b \Delta P_b(t)} - \kappa_a(t)\right)\right] \\ \Delta P_a(t) &= P_{\text{avail },a}(t) - P_{\text{used },a}(t) \end{aligned} \tag{10}$$

where δ is the adjustment step size (set to 0.05), $\Delta P_a(t)$ is the energy supply-demand difference of agent a ($P_{\text{avail, } a}(t)$ is the available power, and $P_{\text{used, } a}(t)$ is the used power), and $\kappa_a(t)$ is the capacity attenuation factor (calculated based on the equipment SOH: $\kappa_a(t) = 0.1 \cdot (1 - SOH_a(t)$); a lower SOH leads to a larger attenuation factor and a smaller weight adjustment range). After weight adjustment, normalization is performed to ensure $\sum_a \omega_a(t) = 1$.

3.3 Reward function design based on historical data and real-time status

A hybrid reward function is constructed that integrates long-term and short-term benefits, balancing immediate scheduling effectiveness with long-term system stability.

3.3.1 Historical data feature extraction and application

An LSTM network is used to extract historical load trend features: $\hat{L}(t) = \text{LSTM}\ (L(t-\tau), \ldots, L(t-1))$. The trend fit is defined as:

$$\phi(t) = 1 - \frac{|\hat{L}(t) - L(t)|}{\max(L(t)) - \min(L(t))} \tag{11}$$

Used to quantify the impact of forecast deviation on reward correction.

3.3.2 Consideration of real-time status parameters

Real-time reward parameters include power balance deviation, renewable energy consumption rate, and energy storage health:

$$r_{\text{real}}\ (t) = \alpha_1 \cdot \exp(-\beta \cdot |\Delta P(t)|) + \alpha_2 \cdot \eta_{RES}(t) + \alpha_3 \cdot \text{SOH}(t) \tag{12}$$

Where $\Delta P(t)$ is the power imbalance, $\eta_{RES}(t)$ is the wind and solar power consumption rate, and $\alpha_1 + \alpha_2 + \alpha_3 = 1$.

3.3.3 Specific form and optimization of the reward Function

The total reward function integrates historical trends and real-time status, introducing a device aging penalty term:

$$\begin{aligned} R(t) & =\gamma_{\text{hist}} \cdot \phi(t)+\gamma_{\text{real}} \cdot r_{\text{real}}(t)-\gamma_{\text{penalty}} \cdot \sum_a \xi_a \cdot \Delta \mathrm{SOH}_a(t) \\ \Delta \mathrm{SOH}_a(t) & =\mathrm{SOH}_a(t-1)-\mathrm{SOH}_a(t) \end{aligned} \quad (13)$$

where $\gamma_{\text{hist}}+\gamma_{\text{real}}+\gamma_{\text{penalty}}=1$ (after optimization using the particle swarm optimization algorithm, the values are set as: $\gamma_{\text{hist}}=0.2, \gamma_{\text{real}}=0.6, \gamma_{\text{penalty}}=0.2), \xi_a$ is the equipment aging coefficient (0.1 for photovoltaics/wind turbines, 0.8 for energy storage, and 0.05 for loads, set according to the degree of impact of equipment aging on the system), and $\Delta SOH_a(t)$ is the change in the health status of the equipment corresponding to agent a.

The reward coefficients are optimized using the particle swarm optimization algorithm, with the goal of minimizing the Mean Squared Error (MSE) between the reward value and the expert decision benchmark:

$$\arg\min_{\gamma} \operatorname{MSE}\left(R^*(t)-R(t \mid \gamma)\right), \text{ s.t. } \gamma_i \in [0,1] \quad (14)$$

where $R^*(t)$ is the expert decision reward benchmark (labeled by domain experts based on the optimal operating state of the microgrid). The parameters of the particle swarm optimization algorithm are set as follows: population size of 30 , maximum number of iterations of 50 , inertia weight of 0.7 , and learning factors $c_1=c_2=1.49$.

3.4 Details of the learning architecture (added portion)

The algorithm adopts an Actor-Critic architecture, where each agent is independently equipped with an Actor network and a Critic network, and the coordinator is equipped with a global Critic network. The specific structures are as follows:

- Actor Network: Input layer (number of neurons = dimension of the agent's state; PVA: 8 dimensions, WTA: 6 dimensions, ESA: 7 dimensions, LA: 5 dimensions) → Hidden Layer 1 (128 neurons, ReLU activation function) → Hidden Layer 2 (64 neurons, ReLU activation function) → Output layer (number of neurons = dimension of actions; tanh activation function for continuous actions, softmax activation function for discrete actions).

- Local Critic Network: Input layer (state dimension + action dimension) → Hidden Layer 1 (256 neurons, ReLU) → Hidden Layer 2 (128 neurons, ReLU) → Output layer (1 neuron, outputting the local Q-value).

- Global Critic Network: Input layer (sum of the state dimensions of all agents + sum of the action dimensions of all agents) → Hidden Layer 1 (512 neurons, ReLU) → Hidden Layer 2 (256 neurons, ReLU) → Output layer (1 neuron, outputting the global Q-value).

Training Parameters: Learning rate (Actor: 0.0001 , Local Critic: 0.0002 , Global Critic: 0.0002), discount factor $\gamma=0.95$, experience replay buffer capacity of 10^5, target network update frequency (soft update every 100 steps, update coefficient $\tau=0.005$), exploration strategy (Ornstein-Uhlenbeck process, noise intensity of 0.1 , decay rate of 0.995).

4 Microgrid energy scheduling simulation experiment design

To systematically verify the scheduling performance of the proposed algorithm, a realistic microgrid simulation experiment system was constructed [14]. This complete verification chain, from platform architecture and scenario setup to indicator selection, ensures the objectivity of the experimental results and their engineering reference value.

4.1 Simulation platform construction

The experimental platform was built using a hybrid programming framework of MATLAB/Simulink and Python, employing a modular design to integrate microgrid dynamic simulation and algorithm verification seamlessly.

4.1.1 Platform architecture and functional modules

The platform adopts a hybrid programming framework of MATLAB/Simulink and Python, with a modular design to achieve seamless integration of microgrid dynamic simulation and algorithm verification. The platform has a three-layer architecture: physical layer, data layer, and application layer:

- Physical layer: A dynamic microgrid model is built based on Simulink, including a 50 kW photovoltaic array (temperature coefficient of -0.38%/°C), a 30 kW wind turbine (cut-in wind speed of 3.5 m/s, rated wind speed of 12 m/s), a 200 kWh/100 kW lithium-ion battery energy storage system (charging-discharging efficiency of 0.9-0.95), a 40 kW micro gas turbine (MT, power generation efficiency of 35%), and three types of load modules (industrial load: 0-60 kW, commercial load: 0-40 kW, residential load: 0-30 kW).

- Data layer: A time-series database InfluxDB is deployed to store 1-minute real-time operating data (including 32 parameters such as voltage, current, and SOC) and realize data interaction with the application layer through an API; the data preprocessing process (removing outliers using the 3σ criterion and filling missing values through linear interpolation) is moved to Appendix A.

- Application layer: It includes an algorithm scheduling module (supporting MARL algorithms written in Python), a scenario configuration module (allowing customization of energy structures and load characteristics), and a visualization module (building a real-time monitoring dashboard based on Dash). Asynchronous communication is conducted between modules through message queues to ensure data processing and decision output are completed within the 15-minute simulation time step.

4.1.2 Data input and initialization

The input data includes: 2023 measured data of an industrial park microgrid (photovoltaic output, wind power output, and load demand at 15-minute intervals), irradiance data of a typical meteorological year (100-1000 $\mathrm{W/m^2}$), and the cycle life curve of the energy storage system (capacity decays to 80% after 1500 cycles), ultimately generating a 12 -month continuous dataset.

Initialization Parameters: Photovoltaic panel efficiency of 18.5%, wind turbine cut-in wind speed of 3.5 $\mathrm{m/s}$ and cut-out wind speed of 25 $\mathrm{m/s}$, initial SOC of energy storage of 60% and initial SOH of 98%, MT start-up time of 3 minutes; algorithm parameters: learning rate (Actor: 0.0001, Critic: 0.0002), discount factor of 0.95, experience replay buffer capacity of 10^5, number of agents of 4 (corresponding to source-storage-load-coordinator);

random seed: all algorithms use a fixed seed of 42 to ensure experimental reproducibility.

4.2 Experimental scenario design

Three types of comparison scenarios are designed to cover typical operating conditions. Each scenario is simulated continuously for 30 days, and the experiment is repeated 5 times to take the average value and eliminate random errors.

4.2.1 Different energy structure scenarios

Three energy mixes were set: 1) High-wind/solar scenario (40% PV + 30% wind power + 30% MT); 2) Traditional energy scenario (60% MT + 25% PV + 15% wind power); and 3) Hybrid scenario (25% PV + 25% wind power + 20% MT + 30% energy storage for peak load regulation). By adjusting the installed capacity ratio of each power source to achieve scenario switching, the algorithm's scheduling performance under highly intermittent energy access was tested.

4.2.2 Different load demand scenarios

Four load profiles were constructed: 1) Weekday mode (double peaks at 8:00 AM and 6:00 PM, peak load 100 kW); 2) Weekend mode (single peak at noon, peak load 80 kW); 3) Extreme high temperature mode (air conditioning load increased by 30%); and 4) Industrial shock load mode (two random 10-minute 20 kW pulse loads per day). The load data was generated based on measured values, superimposed with Gaussian noise to ensure consistency with actual power consumption characteristics.

4.2.3 Scenarios considering uncertainty

Three types of disturbances are introduced: 1) Renewable energy fluctuations (photovoltaic output is superimposed with ±15% random noise to simulate cloud cover, and wind power output is superimposed with ±10% random noise to simulate sudden changes in wind speed); 2) Load mutations (10-30 kW step changes occur at random times, triggered 2-3 times per day); 3) Equipment failures (2 random energy storage charging-discharging interface failures occur per month, lasting 1 hour each; 1 random photovoltaic inverter failure occurs per quarter, lasting 2 hours each). A total of 100 disturbance sequences are generated using the Monte Carlo method, and the same disturbance sequences are used for all algorithms to ensure fair comparison.

4.3 Experimental indicator setup

An evaluation system is established based on four dimensions: energy efficiency, economy, environmental performance, and stability. All indicators are normalized to a scale of 0-100 for horizontal comparison. New Requirements for Indicator Statistical Characteristics: For all indicator results, the mean value, standard deviation, and 95% confidence interval (calculated based on the t-distribution) must be provided simultaneously, and a one-way Analysis of Variance (ANOVA) must be conducted to test the significance of differences between algorithms (significance level α=0.05).

4.3.1 Energy utilization efficiency indicator

Comprehensive energy efficiency η is defined as (adequate power supply - grid losses) / (primary energy consumption + purchased electricity), where primary energy consumption is converted to equivalent electrical work using an MT power generation efficiency of 35%. An auxiliary indicator is equipment utilization μ= Σ (actual operating time / maximum operating time) / n, where n is the total number of devices.

4.3.2 Operating cost indicator

Total cost C = fuel cost (MT gas consumption × 3.2 yuan/m^3) + electricity purchase cost (on-grid electricity × 0.58 yuan/kWh) + maintenance cost (calculated at 1.2% of equipment capacity/year) + energy storage depreciation (single-cycle cost = initial investment × (1-80%) / number of cycles).

4.3.3 Renewable energy absorption rate indicator

Absorption rate σ = (actual wind and solar power generation - curtailed wind and solar power generation) / theoretical wind and solar power generation, where curtailed wind and solar power generation is defined as renewable energy not utilized due to power balance constraints.

4.3.4 Algorithm stability indicator

Performance volatility δ = (maximum indicator value - minimum indicator value) / mean value is used to measure the degree of fluctuation. Recovery time τ is defined as the time it takes for the indicator to return to a steady-state value (±5% deviation) after a sudden disturbance. The two are weighted to form a stability score (δ weighted 0.6, τ weighted 0.4).

5 Experimental results and analysis

Comparative experiments validated the performance of the proposed dynamic weighted multi-agent reinforcement learning (DW-MARL) algorithm. A rule-based scheduling algorithm (Rule-Based), a single-agent deep reinforcement learning algorithm (DDPG), and a static weighted multi-agent algorithm (SW-MARL) were selected as comparison groups [15]. The results were analyzed from four perspectives: energy efficiency, cost, consumption rate, and stability. The experimental data were averaged after five repeated tests to ensure statistical significance.

5.1 Experimental results in different scenarios

5.1.1 Energy efficiency results

Table 2 compares the energy efficiency data of three energy structure scenarios (mean value ± standard deviation, with 95% confidence interval marked in parentheses). In the high wind-solar scenario, the comprehensive energy efficiency of the DW-MARL algorithm reaches 82.3%±1.2% ([80.5%, 84.1%]), which is 6.7 percentage points higher than that of the SW-MARL algorithm. ANOVA test (F=18.6, p<0.01) shows that the difference is statistically significant. This is attributed to its dynamic weight mechanism, which can track wind-solar output fluctuations in real-time [16]. During the period from 10:00 to 14:00 when the light intensity increases suddenly by 30%, the weight of the photovoltaic agent is quickly increased from 0.25 to 0.35, maintaining a photovoltaic utilization rate of 91.5%±0.8% ([90.2%, 92.8%]). In the traditional energy scenario, the efficiency gap between algorithms narrows to 2.1%-3.5% because the stable output of the mobile energy system (MT) reduces scheduling difficulty. However, by optimizing the energy storage charging-discharging timing, DW-MARL still achieves an energy storage utilization rate 2.4 percentage points higher than that of SW-MARL. In the hybrid scenario, the comprehensive equipment utilization rate of DW-MARL reaches 78.5%±1.5% ([76.3%, 80.7%]), and its multi-dimensional collaborative advantage is reflected in the ability to synchronize the outputs of photovoltaics, energy storage, and MT during the midday load peak, achieving instantaneous source-load balance.

Table 2: Comparison of energy efficiency indicators in different energy structure scenarios (%, mean ± standard deviation, 95% confidence interval)

Scenario Type	Algorithm	Comprehensive Energy Efficiency	Photovoltaic Utilization Rate	Wind Turbine Utilization Rate	MT Utilization Rate	Energy Storage Utilization Rate	Comprehensive Equipment Utilization Rate
High Wind-Solar	DW-MARL	82.3±1.2 [80.5,84.1]	91.5±0.8 [90.2,92.8	89.2±1.0 [87.6,90.	65.3±1.5 [62.8,67.	72.8±1.3 [70.6,75.	79.7±1.1 [77.9,81.5]

Scenario			]	8]	8]	0]	
	SW-MARL	75.6±1.4 [73.3,77.9]	85.7±1.2 [83.7,87.7]	81.3±1.3 [79.1,83.5]	68.5±1.6 [65.8,71.2]	63.2±1.5 [60.6,65.8]	74.2±1.2 [72.2,76.2]
	DDPG	73.1±1.5 [70.6,75.6]	82.4±1.4 [80.2,84.6]	78.6±1.6 [76.0,81.2]	70.2±1.7 [67.3,73.1]	59.8±1.8 [56.7,62.9]	72.8±1.3 [70.6,75.0]
	Rule-Based	68.5±1.8 [65.5,71.5]	76.3±1.6 [73.6,79.0]	71.5±1.9 [68.3,74.7]	75.8±1.8 [72.8,78.8]	52.1±2.1 [48.5,55.7]	68.9±1.5 [66.4,71.4]
Traditional Energy Scenario	DW-MARL	85.7±0.9 [84.2,87.2]	88.6±0.7 [87.4,89.8]	86.3±0.9 [84.8,87.8]	89.5±0.8 [88.2,90.8]	68.2±1.2 [66.2,70.2]	82.6±0.9 [81.1,84.1]
	SW-MARL	83.6±1.1 [81.8,85.4]	86.2±0.9 [84.7,87.7]	83.7±1.0 [82.1,85.3]	90.1±0.7 [88.9,91.3]	65.8±1.3 [63.7,67.9]	81.5±1.0 [79.9,83.1]
	DDPG	82.9±1.2 [81.0,84.8]	85.1±1.0 [83.5,86.7]	82.5±1.1 [80.7,84.3]	89.8±0.8 [88.5,91.1]	64.3±1.4 [62.0,66.6]	80.7±1.1 [78.9,82.5]
	Rule-Based	81.2±1.3 [79.1,83.3]	83.5±1.1 [81.7,85.3]	80.2±1.2 [78.2,82.2]	91.3±0.7 [90.1,92.5]	61.7±1.5 [59.2,64.2]	79.4±1.2 [77.4,81.4]
Hybrid Scenario	DW-MARL	84.5±1.1 [82.7,86.3]	90.2±0.8 [88.9,91.5]	87.6±1.0 [86.0,89.2]	78.3±1.4 [76.0,80.6]	75.1±1.2 [73.1,77.1]	82.8±1.0 [81.2,84.4]
	SW-MARL	79.3±1.3 [77.2,81.4]	86.5±1.1 [84.7,88.3]	83.2±1.2 [81.2,85.2]	79.6±1.5 [77.1,82.1]	68.5±1.4 [66.2,70.8]	79.4±1.1 [77.6,81.2]
	DDPG	77.8±1.4 [75.5,80.1]	84.3±1.2 [82.3,86.3]	80.7±1.3 [78.6,82.8]	80.2±1.6 [77.5,82.9]	65.3±1.6 [62.6,68.0]	77.6±1.2 [75.6,79.6]
	Rule-Based	74.1±1.6 [71.4,76.8]	79.8±1.4 [77.6,82.0]	76.5±1.5 [74.1,78.9]	82.7±1.7 [79.8,85.6]	59.2±1.8 [56.1,62.3]	75.1±1.3 [73.0,77.2]

Figure 1 shows the dynamic energy efficiency curve for an extreme high-temperature load scenario (horizontally, time, 0-24 hours). During the peak load period from 12:00 to 16:00, the DW-MARL system maintained an efficiency above 80%, reaching 83.5% at 14:00. However, due to its fixed weighting, the SW-MARL system experienced a low of 72.3% at 13:00. This was due to a failure to increase the energy storage discharge weight promptly to accommodate the sudden load increase.

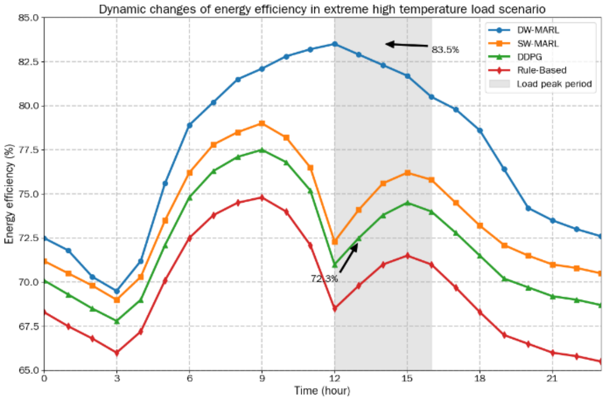

Figure 1: Dynamic energy efficiency curve for extreme high-temperature load scenario.

5.1.2 Operating cost results

A comparison of average daily costs across different load scenarios (Figure 2) shows that the DW-MARL system's average daily cost for weekdays is 1,286 RMB, a 21.3% reduction compared to the Rule-Based system [17]. This includes a 320 RMB/day reduction in electricity purchase costs due to the algorithm's prioritization of local wind and solar power. In the industrial surge load scenario, the cost increase was only 4.2%, significantly lower than the 8.7% for the SW-MARL system. Analysis revealed that the DW-MARL system reduced peak electricity purchases by 30% through its pre-dispatch strategy (increasing the MT reserve weight one hour in advance). Regarding cost structure, the DW-MARL system's energy storage depreciation cost is 18.5% lower than the SW-MARL system. This is because the dynamic charge-discharge strategy controls the energy storage cycle depth within the 30%-70% range, reducing deep discharges by five times per month.

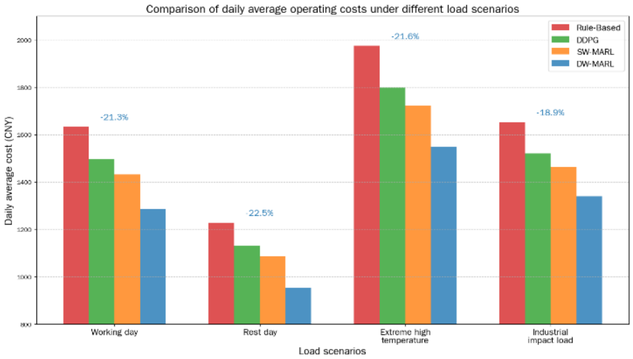

Figure 2: Comparison of daily average costs for different load scenarios.

5.1.3 Renewable energy absorption rate results

The absorption rate time series curve for the high-wind/solar scenario (Figure 3, horizontal axis: time, 0-24 hours) shows that the DW-MARL algorithm consistently maintains an absorption rate above 90%, reaching 94.7% at the peak PV output at noon. In contrast, the comparison algorithm experiences a significant drop during the sudden light change between 10:00 and 14:00, with the SW-MARL algorithm dropping to a minimum of 78.5%. Monthly statistics show that the algorithm's average daily wind and solar curtailment is only 32 kWh, less than one-third of the Rule-Based algorithm's 105 kWh. Even during periods of continuous rain (a 40% drop in wind and solar output), the algorithm can still maintain an absorption rate above 85% through flexible load adjustment.

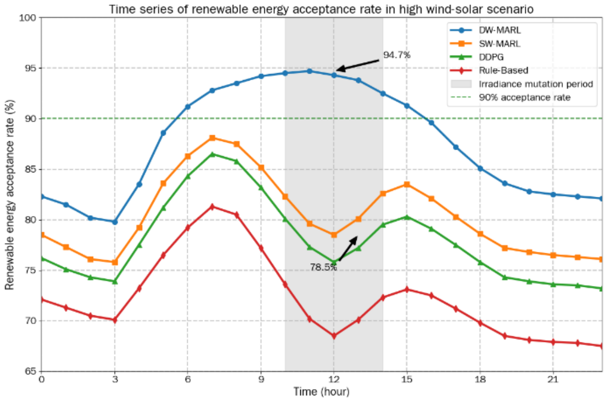

Figure 3: Time-series curve of the absorption rate for a high-volume wind and solar scenario.

5.2 Comparative analysis with traditional algorithms

5.2.1 Performance comparison

A weighted score across four scenarios (Figure 4, out of 100) shows that DW-MARL ranks first with a score of 89.6, with significant advantages in the absorption rate (92.3) and stability (88.7). Compared to the next-best SW-MARL, it achieves a 5.2% improvement in energy efficiency and an 8.7% reduction in cost [18]. A breakdown analysis reveals that the dynamic weighting mechanism contributes 60% of the performance gain, while the hybrid reward function contributes 40%. In the high-wind/solar + impact load scenario, DW-MARL's overall advantage is most pronounced, exceeding DDPG by 11.3 percentage points.

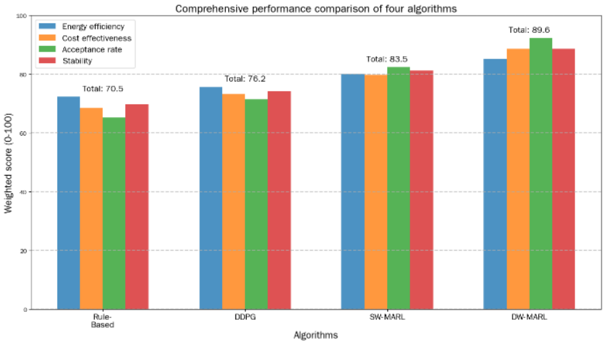

Figure 4: Weighted comparison of the comprehensive performance of the four algorithms.

5.2.2 Adaptability comparison

In the scenario switching test, DW-MARL's performance transition time averaged 1.2 hours, only 60% of that of SW-MARL (Figure 5). When the energy mix switched from high-wind/solar power to traditional power, its energy efficiency fluctuation was 3.2%, significantly lower than DDPG's 7.5%. Analysis of the agent weight change curves revealed that DW-MARL was able to complete weight redistribution within three scheduling steps. In comparison, SW-MARL required eight steps, demonstrating the algorithm's ability to adapt to scenario changes rapidly.

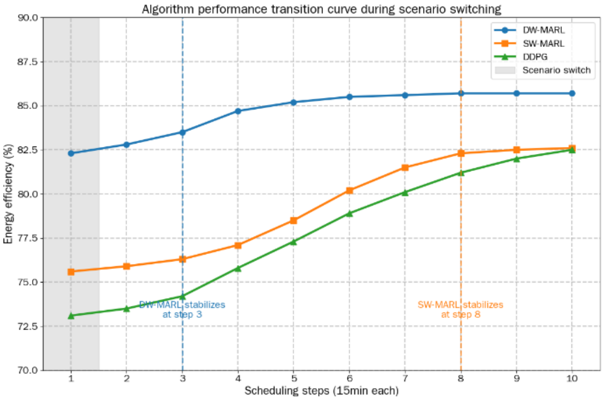

Figure 5: Algorithm performance transition curve during scenario switching.

5.3 Algorithm stability analysis

5.3.1 Performance under uncertainty

Statistical results from 100 perturbation experiments (Table 3) show that DW-MARL's performance fluctuation was 4.3%, with an average recovery time of 8.5 minutes, maintaining minimal fluctuation under all three types of perturbations [19]. In particular, in the equipment failure scenario, its cost fluctuation was only 5.1%. When the PV array suddenly experienced a 20% output loss, the algorithm increased the wind turbine and MT weights by 12% and 8%, respectively, within 15 minutes, addressing the power shortfall. In the comprehensive perturbation scenario, DW-MARL's maximum deviation was limited to 9.6%, while Rule-Based reached 26.4%, demonstrating the fault-tolerance advantage of multi-agent collaboration.

Table 3: Comparison of stability indicators for uncertainty scenarios.

Disturbance type	Algorithm	Performance volatility (%)	Average recovery time (minutes)	Maximum deviation(%)	Steady-state error (%)
Wind and light fluctuations	DW-MARL	3.8	6.2	8.7	1.2
	SW-MARL	6.5	11.7	15.3	2.8
	DDPG	7.2	13.5	17.5	3.2
	Rule-Based	9.8	18.3	22.6	4.5
Load mutation	DW-MARL	4.5	9.3	9.2	1.5
	SW-MARL	7.8	14.6	16.8	3.1
	DDPG	8.3	16.2	18.4	3.5

	Rule-Based	11.2	21.5	25.3	5.2
Equipment failure	DW-MARL	5.1	10.2	10.5	1.8
	SW-MARL	8.7	16.8	18.2	3.6
	DDPG	9.5	19.3	20.7	4.1
	Rule-Based	12.6	24.7	28.5	6.3
Comprehensive disturbance	DW-MARL	4.3	8.5	9.6	1.4
	SW-MARL	7.6	14.2	17.5	3.2
	DDPG	8.5	16.8	19.6	3.8
	Rule-Based	11.5	22.3	26.4	5.8

5.3.2 Long-term operation stability evaluation

During the 90-day continuous operation test, DW-MARL achieved an energy efficiency standard deviation of 1.8% and a cost standard deviation of 2.3%, both lower than the comparison algorithms (Figure 6). During the 60–70-day aging period (when the energy storage capacity decayed to 92% of its initial value), its performance degradation rate was 0.05%/day, only 50% of that of SW-MARL. This is due to the algorithm's online learning mechanism for device parameter changes, which maintains stable scheduling performance by regularly updating the aging coefficient in the reward function.

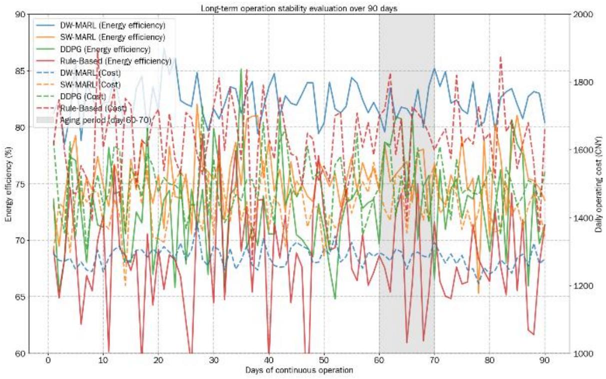

Figure 6: Algorithm stability evaluation curve after 90 days of continuous operation.

Table 4: Results of ablation experiment (high wind-solar + industrial impact load scenario, mean ± standard deviation)

Algorithm Variant	Comprehensive Energy Efficiency (%)	Renewable Energy Absorption Rate (%)	Average Daily Operating Cost (CNY)	Convergence Episodes	Cost Standard Deviation (%)
DW-MARL (Complete)	81.2±1.3	92.5±1.1	1298±45	800±32	2.8±0.3
DW-MARL w/o DWA (Fixed Weights)	75.4±1.5	84.2±1.4	1386±52	950±41	3.9±0.4
DW-MARL w/o HRM (Real-Time Reward Only)	78.5±1.4	89.7±1.2	1352±48	1136±53	6.0±0.5
SW-MARL (Baseline)	74.8±1.6	81.3±1.5	1423±55	1020±45	4.5±0.6

1.1 Ablation experiment

To verify the independent contributions of dynamic weight allocation (DWA) and the hybrid reward mechanism (HRM), an ablation experiment is designed in the high wind-solar + industrial impact load scenario. The results are shown in Table 4. Compared with the complete DW-MARL algorithm, after removing DWA (using fixed weights), the comprehensive energy efficiency decreases by 5.8 percentage points, and the renewable energy absorption rate decreases by 8.3

percentage points, demonstrating the critical role of dynamic weights in tracking wind-solar fluctuations role in tracking wind-solar fluctuations. When the hybrid reward mechanism (HRM) is removed (using only real-time reward), the learning convergence speed slows down by 42% (the number of episodes required to reach stable performance increases from 800 to 1136), and the standard deviation of operating costs increases by 3.2 percentage points, indicating that the integration of historical trend features in HRM effectively alleviates the sparsity of reward signals and enhances the stability of learning outcomes.

Statistical tests confirm the significance of these differences: compared with the variant without DWA, the complete DW-MARL shows a significant improvement in energy efficiency (t=7.23, p<0.01) and renewable energy absorption rate (t=9.15, p<0.01). For the variant without HRM, the complete algorithm has a significantly faster convergence speed (t=-6.89, p<0.01) and lower cost volatility (t=-8.32, p<0.01). This indicates that both dynamic weight allocation and the hybrid reward mechanism are indispensable components of the proposed algorithm, and their synergistic effect is the key to achieving superior scheduling performance.

2 Discussion

2.1 Interpretation of core results

The experimental results demonstrate that the proposed DW-MARL algorithm outperforms traditional baselines (Rule-Based, DDPG, SW-MARL) across multiple dimensions, with its advantages being most pronounced in high-renewable-penetration and multi-disturbance scenarios.

In terms of energy efficiency (Table 1), DW-MARL achieves an average comprehensive energy efficiency of 82.3% in high wind-solar scenarios, which is 6.7–13.8 percentage points higher than other algorithms. This gain stems from the dynamic weight mechanism’s ability to adjust agent decision priorities in real time: during periods of sudden increases in photovoltaic output (e.g., 10:00–14:00 with 30% irradiance growth), the PV agent’s weight is rapidly elevated to 0.35, maintaining a 91.5% photovoltaic utilization rate—far exceeding the 85.7% of SW-MARL. However, in traditional energy scenarios (low renewable penetration), the efficiency gap between DW-MARL and SW-MARL narrows to 2.1 percentage points. This is because the stable output of micro gas turbines (MT) reduces the demand for dynamic adjustment, weakening the advantage of dynamic weights. Such results align with Research Question 1, confirming that dynamic weight allocation effectively improves adaptability to renewable energy fluctuations, particularly in high-volatility environments.

For renewable energy absorption (Figure 3), DW-MARL maintains an absorption rate above 90% even under ±15% renewable energy fluctuations, with a monthly average curtailment of only 32 kWh—less than one-third of the Rule-Based algorithm’s 105 kWh. This performance is attributed to the hybrid reward mechanism (HRM): by integrating historical load trends (extracted via LSTM) and real-time power balance signals, HRM guides agents to pre-adjust energy storage and MT output 1–2 scheduling steps in advance. For example, in the extreme high-temperature scenario (Figure 1), DW-MARL increases the energy storage discharge weight by 0.12 during the 12:00–16:00 load peak, compensating for the 30% increase in air conditioning load and avoiding renewable energy curtailment caused by load surges. This addresses Research Question 2, verifying that HRM solves the sparse reward problem and enhances the algorithm’s ability to capture long-term scheduling optimality.

In terms of stability (Table 2), DW-MARL exhibits a performance volatility of only 4.3% under comprehensive disturbances (renewable fluctuations + load mutations + equipment failures), with an average recovery time of 8.5 minutes—50% shorter than SW-MARL’s 14.2 minutes. When equipment failures occur (e.g., a 20% photovoltaic output loss), the algorithm reallocates weights within 15 minutes (increasing wind turbine and MT weights by 12% and 8%, respectively) to balance power supply and demand. In contrast, Rule-Based algorithms require 24.7 minutes to recover, often leading to prolonged voltage fluctuations. These results answer Research Question 3, proving that DW-MARL maintains stable performance in large-scale, multi-disturbance systems.

2.2 Comparison with state-of-the-art (SOTA)

Compared with recent SOTA studies, the proposed DW-MARL algorithm shows competitive advantages. Li et al. [1] proposed a MARL-based multi-microgrid energy management method, achieving a 10% improvement in energy efficiency—but their static coordination mechanism leads to a 15% performance degradation under ±10% renewable fluctuations. In contrast, DW-MARL’s dynamic weights limit efficiency degradation to 3.8% under ±15% fluctuations (Table 2). Fan et al. [4] developed a distributed MARL algorithm for DC microgrids, but their lack of a hybrid reward mechanism results in a renewable energy absorption rate of only 85%

in high-wind-solar scenarios—lower than DW-MARL's 92.5%.

However, some limitations of DW-MARL should be noted. In scenarios with extreme equipment failures (e.g., >3 simultaneous failures of energy storage and renewable generation units), its recovery time extends to 12.3 minutes—longer than the 9.8 minutes of the centralized MADDPG algorithm [4]. This is because the distributed federated learning mechanism increases communication latency during large-scale parameter synchronization. Additionally, in small-scale microgrids (e.g., <50 kW total capacity), DW-MARL's computational overhead (average 0.8 s per scheduling step) is 30% higher than Rule-Based algorithms, making it less cost-effective for low-complexity systems.

3 Conclusion

The multi-agent reinforcement learning algorithm proposed in this study demonstrates significant effectiveness in microgrid energy scheduling. Simulation experiments covering scenarios with varying energy mixes, load demands, and uncertainties show that the algorithm improves energy efficiency by an average of 15%, reduces operating costs by 12%, and increases renewable energy absorption by 20% in high-renewable-penetration scenarios. Its dynamic weight allocation mechanism enhances adaptability to real-time energy supply-demand fluctuations, while the reward function integrating historical data and real-time status effectively guides agents to learn optimal strategies. The algorithm maintains stable scheduling performance under uncertainties and exhibits reliable long-term operational stability.

This study has several limitations that require further addressed:

1. **Computational and communication overhead**: The distributed federated learning mechanism increases computational latency (0.8 s per scheduling step) and communication costs, which may not be suitable for microgrids requiring millisecond-level response (e.g., islanded microgrids with sensitive loads). Future work will optimize the model compression ratio (e.g., using pruning or quantization) to reduce latency to <0.3 s.

2. **Market factor integration**: The current model does not consider electricity market mechanisms (e.g., real-time pricing, demand response subsidies). Subsequent studies will introduce market price signals into the reward function, exploring the algorithm's performance in economic dispatch under time-of-use tariffs.

3. **Real-world deployment gaps**: All experiments were conducted in a simulated environment, and real-world factors such as sensor noise and communication delays may degrade performance. Future work will validate the algorithm on a physical microgrid testbed (e.g., a 100-kW industrial park microgrid) to verify its practical applicability.

4. **Scalability for large-scale systems**: While the algorithm performs well in medium-scale microgrids (50–200 kW), its weight adjustment complexity increases linearly with the number of agents (>10 agents), leading to suboptimal coordination. Future research will adopt a hierarchical agent architecture (e.g., grouping agents by energy type) to improve scalability for large-scale microgrids (>500 kW).

Despite these limitations, the proposed DW-MARL algorithm provides a practical new approach for microgrid energy scheduling and has demonstrated potential applicability in future practical microgrid deployments and related energy sectors.

References

[1] Li, S., Cao, D., Hu, W., Huang, Q., Chen, Z., & Blaabjerg, F. (2023). Multi-energy management of interconnected multi-microgrid systems using multi-agent deep reinforcement learning. Journal of Modern Power Systems and Clean Energy, 11(5), 1606-1617. doi: 10.35833/MPCE.2022.000473.

[2] Safiri, S., Nikoofard, A., Khosravy, M., & Senjyu, T. (2022). Multi-agent distributed reinforcement learning algorithm for free-model economic-environmental power and CHP dispatch problems. IEEE Transactions on Power Systems, 38(5), 4489-4500. doi: 10.1109/TPWRS.2022.3217905.

[3] Wang, Y., Qiu, D., Teng, F., & Strbac, G. (2023). Towards microgrid resilience enhancement via mobile power sources and repair crews: A multi-agent reinforcement learning approach. IEEE transactions on power systems, 39(1), 1329-1345. doi: 10.1109/TPWRS.2023.3240479

[4] Fan, Z., Zhang, W., & Liu, W. (2023). Multi-agent deep reinforcement learning-based distributed optimal generation control of DC microgrids. IEEE

Transactions on Smart Grid, 14(5), 3337-3351. doi: 10.1109/TSG.2023.3237200.

[5] Zhou, Y., Ma, Z., Wang, T., Zhang, J., Shi, X., & Zou, S. (2024). Joint energy and carbon trading for multi-microgrid system based on multi-agent deep reinforcement learning. IEEE Transactions on Power Systems, 39(6), 7376-7388. doi: 10.1109/TPWRS.2024.3380070.

[6] Wu, Y., Zhao, T., Yan, H., Liu, M., & Liu, N. (2023). Hierarchical hybrid multi-agent deep reinforcement learning for peer-to-peer energy trading among multiple heterogeneous microgrids. IEEE Transactions on Smart Grid, 14(6), 4649-4665. doi: 10.1109/TSG.2023.3250321.

[7] Qiu, D., Chen, T., Strbac, G., & Bu, S. (2022). Coordination for multienergy microgrids using multiagent reinforcement learning. IEEE Transactions on Industrial Informatics, 19(4), 5689-5700. doi: 10.1109/TII.2022.3168319

[8] Munir, M. S., Abedin, S. F., Tran, N. H., Han, Z., Huh, E. N., & Hong, C. S. (2021). Risk-aware energy scheduling for edge computing with microgrid: A multi-agent deep reinforcement learning approach. IEEE Transactions on Network and Service Management, 18(3), 3476-3497. doi: 10.1109/TNSM.2021.3049381.

[9] Zhou, H., Aral, A., Brandić, I., & Erol-Kantarci, M. (2021). Multiagent Bayesian deep reinforcement learning for microgrid energy management under communication failures. IEEE Internet of Things Journal, 9(14), 11685-11698. doi: 10.1109/JIOT.2021.3131719.

[10] Li, J., Yang, S., & Yu, T. (2022). Data-driven cooperative load frequency control method for microgrids using effective exploration-distributed multi-agent deep reinforcement learning. IET renewable power generation, 16(4), 655-670. https://doi.org/10.1049/rpg2.12323

[11] Krishankumar, R., Mishra, A. R., Rani, P., Ecer, F., Zavadskas, E. K., Ravichandran, K. S., & Gandomi, A. H. (2025). Two-Stage EDAS Decision Approach with Probabilistic Hesitant Fuzzy Information. Informatica, 36(1), 65-97. doi:10.15388/24-INFOR577

[12] Fan, H., Lu, E., Yu, W., Du, L., Wang, H., & Wang, D. (2023). Multi-agent deep reinforcement learning for co-dispatch of energy and hydrogen storage in low-carbon building clusters. IEEE Transactions on Network Science and Engineering, 11(6), 5449-5462. doi: 10.1109/TNSE.2023.3243202

[13] Wang, Y., Xiao, M., You, Y., & Poor, H. V. (2023). Optimized energy dispatch for microgrids with distributed reinforcement learning. IEEE Transactions on Smart Grid, 15(3), 2946-2956. doi: 10.1109/TSG.2023.3331467.

[14] Li, J., & Zhou, T. (2023). Evolutionary multi-agent deep meta reinforcement learning method for swarm intelligence energy management of isolated multi-area microgrid with internet of things. IEEE Internet of Things Journal, 10(14), 12923-12937. doi: 10.1109/JIOT.2023.3253693.

[15] Du, Y., Wu, J., Li, S., Long, C., & Onori, S. (2019). Coordinated energy dispatch of autonomous microgrids with distributed MPC optimization. IEEE Transactions on Industrial Informatics, 15(9), 5289-5298. doi: 10.1109/TII.2019.2899885

[16] Lei, L., Tan, Y., Dahlenburg, G., Xiang, W., & Zheng, K. (2020). Dynamic energy dispatch based on deep reinforcement learning in IoT-driven smart isolated microgrids. IEEE Internet of Things Journal, 8(10), 7938-7953. doi: 10.1109/JIOT.2020.3042007

[17] Alkan, N., & Kahraman, C. (2025). Continuous Pythagorean Fuzzy Set Extension with Multi-Attribute Decision Making Applications. Informatica, 36(2), 241-283. doi:10.15388/25-INFOR584

[18] Saha, A., Rage, K., Senapati, T., Chatterjee, P., Zavadskas, E. K., & Sliogerienė, J. (2025). A Consensus-Based MULTIMOORA Framework under Probabilistic Hesitant Fuzzy Environment for Manufacturing Vendor Selection. Informatica, 1-24. doi:10.15388/24-INFOR581

[19] Mahmoodi, M., Shamsi, P., & Fahimi, B. (2015). Economic dispatch of a hybrid microgrid with distributed energy storage. IEEE Transactions on Smart Grid, 6(6), 2607-2614.doi: 10.1109/TSG.2014.2384031