An Assessment of Multistage Reward Function Design for Deep Reinforcement Learning-Based Microgrid Energy Management

Hui Hwang Goh, *Senior Member, IEEE*, Yifeng Huang, Chee Shen Lim, *Senior Member, IEEE*, Dongdong Zhang, *Member, IEEE*, Hui Liu, *Senior Member, IEEE*, Wei Dai, *Member, IEEE*, Tonni Agustiono Kurniawan, and Saifur Rahman, *Life Fellow, IEEE*

***Abstract*—Reinforcement learning based energy management strategy has been an active research subject in the past few years. Different from the baseline reward function (BRF), the work proposes and investigates a multi-stage reward mechanism (MSRM) that scores the agent's step and final performance during training and returns it to the agent in real time as a reward. MSRM will also improve the agent's training through expert intervention which aims to prevent the agent from being trapped in sub-optimal strategies. The energy management performance considered by MSRM-based algorithm includes the energy balance, economic cost, and reliability. The reward function is assessed in conjunction with two deep reinforcement learning algorithms: double deep *Q*-learning network (DDQN) and policy gradient (PG). Upon benchmarking with BRF, the numerical simulation shows that MSRM tends to improve the convergence characteristic, reduce the explained variance, and reduce the tendency of the agent being trapped in suboptimal strategies. In addition, the methods have been assessed with MPC-based energy management strategies in terms of relative cost, self-balancing rate, and computational time. The assessment concludes that, in the given context, PG-MSRM has the best overall performance.**

***Index Terms*—Microgrid energy management, deep reinforcement learning, reward function, optimal scheduling.**

Manuscript received 11 August 2021; revised 16 January 2022; accepted 28 May 2022. Date of publication 1 June 2022; date of current version 21 October 2022. This work was supported in part by the Guangxi University under Grant A30 20051008, and in part by the National Key Research and Development Program of China under Grant 2019YFE0118000. Paper no. TSG-01281-2021. *(Corresponding author: Hui Hwang Goh.)*

Hui Hwang Goh, Yifeng Huang, Dongdong Zhang, Hui Liu, and Wei Dai are with the School of Electrical Engineering, Guangxi University, Nanning 530004, China. (e-mail: hhgoh@gxu.edu.cn; 1912391016@st.gxu.edu.cn; dongdongzhang@gxu.edu.cn; hughlh@gxu.edu.cn; weidai@gxu.edu.cn).

Chee Shen Lim is with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215131, China (e-mail: cheeshen.lim@xjtlu.edu.cn).

Tonni Agustiono Kurniawan is with the Key Laboratory of the Coastal and Wetland Ecosystems, Ministry of Education, College of the Environment and Ecology, Xiamen University, Xiamen 361102, Fujian, China (e-mail: tonni@xmu.edu.cn).

Saifur Rahman is with the Department of Advanced Research Institute, Virginia Polytechnic Institute and State University, Arlington, VA 22203 USA (e-mail: srahman@vt.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2022.3179567.

Digital Object Identifier 10.1109/TSG.2022.3179567

I. Introduction

MICROGRID infrastructure can facilitate the integration of dispersed energy resources such as wind turbines (WT), photovoltaic (PV), diesel generator, and energy storage system (ESS), as well as controllable loads into future power grids. This contributes to the goal of reaching carbon neutrality or net-zero emission in the society. A microgrid can operate autonomously in the grid-connected mode or the islanded mode, depending on various technical and economic factors [1]. Since its inception in 2003 [2], [3], many microgrid testbeds have been built to provide electrical (and thermal) energy to a variety of populations, including rural houses, public, institutional, industrial, military, and outlying locations, as well as smaller islands and villages that are not connected to the main grid. Most of the renewable energy resources, such as photovoltaic and wind, are weather dependent and are therefore of intermittent nature, requiring the use of intelligent energy management system (EMS). EMS manages and dispatches distributed energy resources (including loads) to achieve energy balance within the microgrid [4].

In the last decade, dynamic programming-based energy management strategy has been the focus at large in the research communities. For example, a model predictive control (MPC) algorithm for real-time energy management based on offline optimal solution was proposed in [5]. In [6], an energy storage management system based on MPC which adapts to the changes of PV output, was proposed and investigated. In most of the model-based solutions, deterministic microgrid models are required, and they are often simplified in numerous ways to reduce the computational complexity. Reference [7] models the online energy management problem as a stochastic optimal power flow problem, taking into account the distribution network's power flow and operational limits. In [8], an energy management incorporating approximation in the dynamic programming model and deep neural network was proposed and investigated. It is worth noting that most of the aforementioned methods require a prior knowledge of the mathematical model of the microgrid network, leading to two known disadvantages [9]: given the diverse ranges of distributed energy resources, it is a known challenge among classical EMS to establish accurate mathematical models; and,

in the case of partially quantifiable system, iterative algorithms are inevitably required to establish the model. However, since convergence is usually not guaranteed by these algorithms, their real-time implementation is rather limited.

In the past few years, the use of deep reinforcement learning (DRL) to overcome the challenges faced by the model-based techniques is gaining research attention, especially during the aftermath of AlphaGo's victory over human chess players [10]. In essence, DRL algorithm is a model-free machine learning technique that is built based on the fundamental concept of sequential decision making, which is accomplished through continuous interaction between the decision makers (agents) and the environment. The agent constantly monitors the environment, takes action, and develops awareness in order to accomplish its objectives in real-time setting [11]. DRL-based approaches eliminate the need of prior modeling of the plant, but instead, it uses iterative calculation, normally through a large amount of data, to build the agent's optimal control/decision strategy. DRL has been used in a variety of applications to date, including video games, robot motion control, self-guided automation, and intelligent human-machine interaction [12].

As far as microgrid application is concerned, DRL algorithms have been studied in conjunction with various optimization of energy storage and dispatch management [13]–[17]. This learning-based technology trend is expected to accelerate the adoption of microgrid technology as they inherently consider the stochastic nature of the renewable resources and fossil-fuel-driven electricity prices. Reference [18] proposes an energy trading algorithm for scheduling the microgrids based on future projections. Deep reinforcement learning is used to determine the energy transaction between the microgrid and the power plants. It was shown that the combination of prediction and decision-making can lead to improved convergence time. Reference [19] proposes a hybrid optimization technique using an artificial neural network (ANN) and Q-learning for a home energy management scheduling. Another home energy management framework is studied in [20], in which the framework merges extreme learning machine and multi-agent Q-learning. The technique places a high premium on prediction engagement, as a result, the forecast could become a hindrance to developing a real-time dispatch strategy. In [21], a smart microgrid dynamic energy management system is developed through the design of a reinforcement learning framework that continuously analyzes current demand and generation data in order to generate real-time dispatch commands. In [22], a dual-network approach based on deep Q-network is proposed for the scheduling of real-time charging or discharging of electric vehicles. On the other hand, a double deep Q-learning algorithm with priority replay strategy and dual structure is studied in [23] for real-time dispatch of community battery energy storages. From previous works, it can be established that in most cases the performance of DRL policies depend rather significantly on the reward function design.

In DRL-based prior arts, the microgrid EMS problem has been solved primarily through the use of incentives/rewards of operating cost and power balance. Intuitively, in the context of agent training, the quality of the reward signal will be affected by: (i) the hard constraints due to the size and variety of dispatchable DERs; and, (ii) the range/variety of the available training episodes (note: an episode is a complete play of the agent interacting with the environment). For example, in an episode where the renewable energy is inherently scarce, even with the maximum output from the size-limited ESS, additional power imports from the main grid are required to ensure an uninterrupted supply. Nevertheless, if the reward to the agent is based only on the running cost and self-balancing rate, under the above scenario, the reward related to the ESS will not contribute to the "on-going" training process. This is a sparse reward problem known to DRL community. This problem typically causes the training to converge more slowly and may even lead to a suboptimal agent/strategy. In the context of classical, forecasting-based optimal energy management, apart from the cost function design, the accuracy of the forecasting techniques will normally dictate the performance of the energy management. Similarly, in the present context, an effective design of the reward function is crucial to produce an accurate and effective scheduling strategy.

This paper proposes a new multi-stage reward mechanism (MSRM) for deep reinforcement learning algorithm to overcome aforementioned problems. First, MSRM comprising step reward, final reward, and external expert intervention are designed. Step and final rewards account for energy cost, power balance, and system reliability of the microgrid. The external expert intervention is designed with microgrid-related conditions to limit the explorable action space during training. In additional, DRL based on baseline reward function (BRF) is added for direct comparison. In terms of DRL algorithms, two of the common algorithms are used here: deep double Q-network (DDQN) and policy gradient (PG). DDQN is well known among the practitioners to overcome the common overestimation problem in DQN [24]. PG, on the other hand, has the advantages of simplicity and convergence [25], and being more effective for high-dimensional action spaces [26]. Some notable PG examples include the high-dimensional robot control problem [27], and large-scale microgrid [28]–[31]. It is also worth noting that more advanced DRL agents, such as deep deterministic policy gradient (DDPG) and actor-critic (AC) [26], [32], which are essentially advanced versions of PG and DQN, are available [33]. The four DRL algorithms (DDQN-MSRM, DDQN-BRF, PG-MSRM, and PG-BRF) are assessed in terms of convergence capability (i.e., reward-episode performance, explained variance) and detailed aspects of microgrid energy management (i.e., battery SOC and power, cost, and CO_2 reduction). In addition, these algorithms are benchmarked against the existing methods (model predictive control with 8/12 moving horizon) in terms of cost, self-balancing rate, and computational complexity.

The remainder of this paper is organized as follows. Section II describes the microgrid structure. Section III discusses the energy dispatch strategy based on Markov Decision Process. Section IV illustrates MSRM and BRF. Section V states the fundamental forms of DDQN and PG. Section VI summarizes and discusses the two-part assessment: comparison among the four DRL algorithms, and the benchmarking

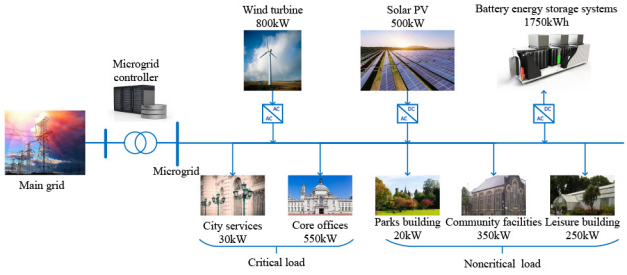


Fig. 1. Example of a community microgrid.

with the energy management based on rolling horizon model predictive control. Lastly, Section VII concludes the paper.

II. Microgrid Structure

Fig. 1 depicts a grid-connected community microgrid that is comprised of wind generation system, solar PV generation system, battery energy storage system, critical load (CL), and noncritical load (NCL). The microgrid operates in grid-connected mode and participates in the real-time electricity market. The daily dispatch is divided into T time steps, indexed by $\{1, 2, \ldots, T\}$. The interval of each time step is $\Delta t = 24/T$. The composition of the microgrid is described in the following subsections.

Active power outputs of the wind and PV generation are known respectively as P_t^{WT} and P_t^{PV}. The training data corresponds to the actual production data of wind and solar plant.

The microgrid contains two types of loads: CLs and NCLs. Training data of CL and NCL correspond to the actual data of the Cardiff Council community in Wales, UK [34]. In the event of the main grid disruption, the microgrid will switch from the grid-connected mode to the islanded mode. To ensure a continued supply to CL, the power dispatch to NCL may be shed accordingly. The loads' total active power P_t^L is calculated using

$$P_t^L = P_t^{CL} + P_t^{NCL} \tag{1}$$

where P_t^{CL} and P_t^{NCL} are the active power demand of CL and NCL load. This distinction can better determine the configuration of energy storage capacity. When the main network is unable to supply power, the noncritical load can be curtailed, and energy storage ensures that the critical load can continue to operate for a longer period of time.

Two types of energy storage system (ESS) are considered in this work: high-power-density ESS (HPE) and high-capacity ESS (HCE). HCE has a high capacity and can be used for long-term power generation. HPE has a high discharge rate, which enables it to be used to meet short-term peak electricity consumption. The battery size is designed to power the CL for at least 3 hours. At each time step t, the charging or discharging power P_t^E and the state of current of the ESS Φ_t^E are constrained by eqns. (2)–(4).

$$0 \le P_t^E \le P_{\max}^E \tag{2}$$

$$\Phi_{\min}^E \le \Phi_t^E \le \Phi_{\max}^E \tag{3}$$

$$\Phi_{t+1}^E = \Phi_t^E(1 - \sigma_t) + I_t^E \cdot \Delta t \cdot \eta_t / B_E \tag{4}$$

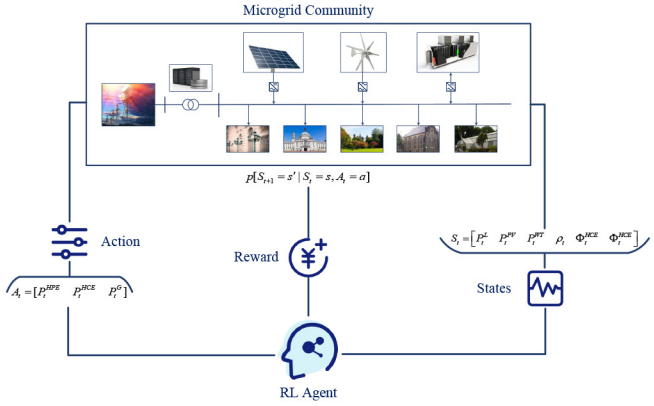


Fig. 2. Markov decision process for microgrid energy management.

where $P_{\max}^E$ is the maximum charging or discharging power, $E = \{HPE, HCE\}$, $\Phi_{\max}^E$ and $\Phi_{\min}^E$ are the maximum and minimum energy levels of the ESS, $\sigma_t \in [0, 1]$ is the self-discharge rate, $\eta_t \in [0, 1]$ is the charging or discharging efficiency, and I_t^E is the charging (positive) or discharging (negative) current.

The power exchanged between the microgrid and the main grid P_t^G should be limited:

$$-P_{\max}^G \le P_t^G \le P_{\max}^G \tag{5}$$

where $P_{\max}^G$ is the maximum active power that can be imported from or exported to the main grid.

The cost of purchasing active power from the main grid at time step t is computed by

$$Cost_t^{Grid} = P_t^G \cdot \rho_t \cdot \Delta t \tag{6}$$

where ρ_t is the real-time electricity price at time step t.

III. Markov Decision Process-Based Energy Dispatching Strategy

As illustrated in Fig. 2, this study models the microgrid energy management process as a Markov decision process (MDP). In general, MDP takes into account the stochastic nature of the renewable energy sources, real-time electricity price changes, and the state of charge of the ESS to determine the system state and the corresponding energy dispatches. An MDP can be described by the tuple $\langle S, A, P, R \rangle$, which is elaborated in the following subsections.

A. State Space

The state at time step t is described as $S_t = (P_t^L, P_t^{PV}, P_t^{WT}, \rho_t, \Phi_t^{HPE}, \Phi_t^{HCE})$ which consists of the following information: the total active power loads P_t^L, the PV active power output P_t^{PV}, the active power output of wind turbines P_t^{WT}, the real-time electricity price ρ_t, the state of charge in the HPE Φ_t^{HPE}, and the state of charge in the HCE Φ_t^{HCE}.

B. Action Space

The microgrid dispatch signals for time step t is described as $A_t = (P_t^{HPE}, P_t^{HCE}, P_t^G)$, which includes the HPE active power P_t^{HPE}, the HCE active power P_t^{HCE}, and the active power exchange with the main grid P_t^G.

C. State Transition Probability

In MDP, the agent will execute action A_t according to the current state S_t of the environment at time step t, and state S_t will transition to S_{t+1}. The state transition probability p describes the probability of transition from state $s \in S$ to $s' \in S$ given an action a:

$$p^a_{ss'} = p\big[S_{t+1} = s' | S_t = s, A_t = a\big]. \tag{7}$$

D. Reward Function

The reward function consists of three parts: self-balancing rate $R_t^{self-balan}$; reliability rate R_t^{reliab}; the cost of electricity transaction with the main grid $Cost_t^{Grid}$ (as defined in (6)). The self-balancing rate and reliability rate are defined in what follows.

The proportion of grid-connected microgrids that rely on distributed energy resources reflects the microgrid's power supply capacity and degree of grid dependence [35]. The load ratio that a grid-connected microgrid can supply entirely on its own during a given period is referred to as the self-balancing rate, which can be modeled as in (8). When the self-balancing rate reaches 1, it indicates that the entire load demand is supplied by energy sources within. It then reduces the carbon dioxide emissions $C_t^{reduction}$ which can be calculated using (9).

$$R_t^{self-balan} = \frac{P_t^{self}}{P_t^{total}} = 1 - \frac{P_t^{grid-to-load}}{P_t^{total}} \tag{8}$$

$$C_t^{reduction} = \tau_{elec} \cdot P_t^{self} \tag{9}$$

where P_t^{self} is the power supplied by distributed energy resources, P_t^{total} includes user load and battery charging power, $P_t^{grid-to-load}$ is the power supplied from the main grid, $C_t^{reduction}$ is the emission reduction achieved, τ_{elec} is gaseous fuel conversion factor provided by the British government [36].

The reliability rate is defined as the ratio of the total power supply to the total load demand, as shown in (10).

$$R_t^{reliab} = \frac{P_t^{PV} + P_t^{WT} + P_t^{HPE} + P_t^{HCE} + P_t^{G}}{P_t^{L}} \tag{10}$$

Then, the reward function of microgrid energy management at time step t is modeled in (11) as.

$$Reward(S_t, A_t) = f(R_t^{self-balan}, R_t^{reliab}, Cost_t^{Grid}) \tag{11}$$

where $f(\cdot)$ is an multi-stage evaluation mechanism based on $R_t^{self-balan}$, R_t^{reliab}, and $Cost_t^{Grid}$.

E. Modeling Markov Decision Process

The final objective of an MDP is to establish an optimal policy π^* that maximizes the expected total reward for the T-hour horizon:

$$\upsilon^* = \max_{\pi} \sum_{t=1}^{T} Reward(S_t, A_t) p_\pi(A_t | S_t) \tag{12}$$

where policy π is a decision rule that determines the action for a given state, and $p_\pi(A_t|S_t)$ is the action probability specified by the policy π in each state.

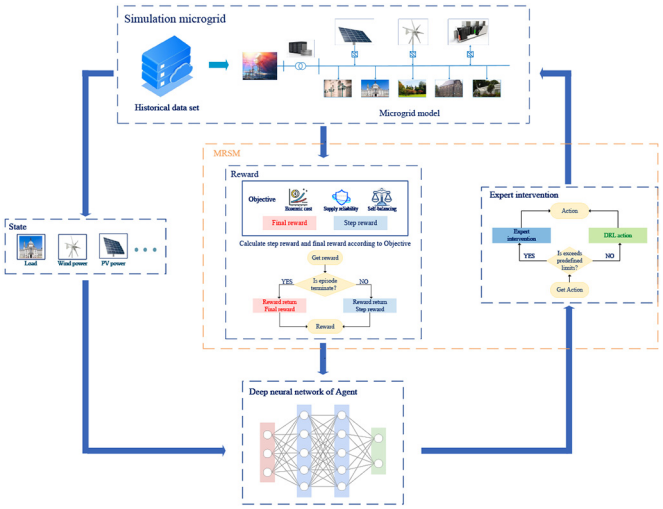


Fig. 3. Reinforcement learning training process using MSRM.

IV. Multi-Stage Reward Mechanism

In typical DRL-based EMS schemes, baseline reward function (BRF) is commonly used to account for the microgrid's operating cost and power balance. The reward is returned to the agent at time step t. The BRF is commonly defined as follows [8], [22], [37]–[39]:

$$r_t = -\left(Cost_t^{Grid} + D_t^{reliab} + D_t^{self-balan}\right) \tag{13}$$

$$Cost_t^{Grid} = \sum_{t}^{T} P_t^G \cdot \rho_t \cdot \Delta t \tag{14}$$

$$D_t^{reliab} = \sum_{t}^{T} \left(\left| P_t^{PV} + P_t^{WT} + P_t^{HPE} + P_t^{HCE} + P_t^{G} - P_t^{L} \right| \right) \tag{15}$$

$$D_t^{self-balan} = \sum_{t}^{T} \left(\left| P_t^{self} - P_t^{total} \right| \right) \tag{16}$$

where, $Cost_t^{Grid}$ is the cost of purchasing active power from the main grid, D_t^{reliab} is active power balance equation, and $D_t^{self-balan}$ is the power equation for the self-supply of the microgrid using distributed energy. The added negative sign signifies that maximizing the rewards (i.e., less negative numerical number) will lead to smaller active power import cost, better active power balance or reliability, and better self-balance. However, as described earlier (and will be shown in the result section), this simple reward function design may lead to sparse reward problems, affecting the quality of the agents [39].

To improve this aspect, a multi-stage reward mechanism (MSRM) that aims to increase the training efficiency and therefore the optimality of the trained learning agent, is proposed. The reward mechanism is comprised of three stages: step reward, final reward, and external expert intervention. The reinforcement learning and training process using MSRM is shown in Fig. 3. What follows illustrates MSRM and its integration with the RL technique. The performance of MSRM will be benchmarked with BRF, as will be elaborated later.

A. Step Reward

Step reward is an instant reward returned to the agent after each step, which can represent an assessment of the current environment. Step reward will return reward according to three indicators , as show in (17).

$$r_t^{step} = \left[\left(1 - \left(\sum_{i=1}^{t} R_i^{self-balan}\right)\Big/ t\right) + \left(\left(\sum_{i=1}^{t} \frac{Cost_i^{Grid}}{Cost_i^*}\right)\Big/ t\right) + \left(1 - \left(\sum_{i=1}^{t} R_i^{reliab}\right)\Big/ t\right)\right] * \zeta_1 \tag{17}$$

$$Cost_t^* = \left(P_t^{CL} + P_t^{NCL} - P_t^{PV} - P_t^{WT}\right) \cdot \rho_t \cdot \Delta t \tag{18}$$

where ζ_1 is a penalty coefficient, and the choice of ζ_1 coefficient is obtained through trail-and-error [15]. The empirically determined value is $\zeta_1 = -10$ (note: negated value is required here to make sure that the step reward can comply with return maximization). The first term of (17) reflects the self-balancing rate of the microgrid from time step 1 to t. The second term of (17) represents the degree of savings in operating costs from time step 1 to t, where $Cost_t^*$ is the electricity transaction cost without energy storage system power supply, as defined in (18). The third term of (17) reflects the reliability rate from time step 1 to t. Note that the purpose of introducing the denominator in the second term is to ensure that its reward value is of the same order as the other two terms, which are in the range of 0 to 1. The denominator is explicitly constrained to be non-zero; alternatively, to achieve the same effect, one can separate the second term from r_t^{step} and introduce a new weighting factor ζ to scale its value range.

B. Final Reward

Final reward keeps track of the environmental changes during each time step and then assesses the agent's behavior during the time step. The final reward will be return rewards to the agent at the conclusion of each episode, giving the agent a comprehensive evaluation with the aim of improving the agent's pursuit of long-term benefits. Similar to the reward function defined in Section III-D, the final reward is also comprised of three components. At the end of an episode, the final reward $r_{final} = \sum_{t=1}^{T} r_{t1} + r_{t2} + r_{t3}$ is returned to the agent, as described below.

The first final reward component evaluates the contribution of the agent's actions to the grid self-balancing rate (defined in (8)):

$$r_{t1} = \begin{cases} \left(1 + \frac{P_t^{E\_discharge}}{P_t^{total}}\right) \cdot \zeta_2, & R_t^{self-balan} \geq 0.95 \\ \left(\frac{P_t^{E\_discharge}}{P_t^{total}}\right) \cdot \zeta_2, & R_t^{self-balan} < 0.95 \end{cases} \tag{19}$$

where $P_t^{E\_discharge}$ is the battery's discharge capacity at the interval t of each time step. $R_t^{self-balan}$ and P_t^{total} have been defined above. ζ_2 is a reward coefficient that is set empirically to 20, and r_{t1} is proportional to the battery output P_t^E. When $R_t^{self-balan} \geq 0.95$, this final reward component gives artificially higher rewards (i.e., with the added reward "1") to the environment when ESS participates more in the scheduling. When $R_t^{self-balan} < 0.95$, this reward component will still give some rewards based on the contribution of the agent's actions.

The second final reward component evaluates the contribution of the agent's actions to the electricity transaction cost (defined in (6)):

$$r_{t2} = \left(1 - \frac{Cost_t^{Grid}}{Cost_t^*}\right) \cdot \zeta_2 \tag{20}$$

Intuitively, this final reward component increases as $Cost_t^{Grid}$ decreases, which means that the cost of electricity decreases.

The third final reward component evaluates the contribution of the agent's actions to the supply reliability (defined in (10)):

$$r_{t3} = \begin{cases} \zeta_2, & R_t^{reliab} \geq 0.95 \\ 0, & R_t^{reliab} < 0.95 \end{cases} \tag{21}$$

where R_t^{reliab} is the reliability rate as already defined in eqn. (10). Effectively, this final reward component exists only when the supply reliability is guaranteed. It is worth highlight that all the final reward components are of the same order, and weighting factor ζ_2 has been added to scale the totaled final reward component against the step reward component. Note also that eqns. (19) and (21) have threshold setting that differentiates them significantly from the step reward. For example, in eqn. (21), the reliability rate must be more than 0.95 to be rewarded with a constant value; otherwise, a zero value. The basis of such setting is that a reliability lower than 0.95 would have a very significant impact on the grid users. In the classical power system, grid reliability typically kept to a very high level (e.g., above 0.99). This value can be readily adjusted by the final user. Intuitively, the "human-perception-based" logic should not apply to the electricity purchase cost, hence the threshold setting does not apply.

C. External Expert Intervention

The external expert intervention stage will assess the current state of the environment and decides explicitly on the outputs during the agent training stage. A basis of designing the "expert" inputs can be through the prior knowledge derived from the classical rule-based or optimization-based energy management techniques. Intuitively, if the state of the environment exceeds predefined limits, external inputs will intervene the learning process through additional rules. In other words, the agent uses the external inputs outside the episodes to modify the policy. The external expert intervention used in this work is summarized as follows:

$$P_{t+1}^E = \begin{cases} P_t^{crtl}, & Soc_{t-1\,to\,t} > 85 \text{ and } R_{t-1\,to\,t}^{self-balan} > 0.9 \\ P_t^{ch}, & Soc_{t-1\,to\,t} < 25 \text{ and } R_{t-1\,to\,t}^{self-balan} < 0.9 \\ P_t^{DRL}, & other \end{cases} \tag{22}$$

$$P_t^{ctrl} = P_t^{CL} + P_t^{NCL} - P_t^{PV} - P_t^{WT} \tag{23}$$

$$P_t^{ch} = U_t^E \cdot I_{ch}^E \tag{24}$$

$$I_{ch}^E = r_{ch} \cdot B_E \tag{25}$$

where P_t^{ctrl} is the ESS power command (under the power control mode), P_t^{ch} is the ESS charging power command (under

Algorithm 1 Multi-Stage Reward Mechanism

Set penalty coefficient ζ_1 and reward coefficient ζ_2
for episode in 1 to M **do**
  Initialize final reward r_{final} and step reward r_t^{step}
  for t in 1 to T **do**
    Observe state S_t and action A_t
    if External expert intervention is met **then**
      P_{t+1}^E is given by expert inputs
    else
      P_{t+1}^E is given by DRL
    end if
    Calculate step reward at every t:$r_t = r_t^{step}$
    Calculate final reward at every t:
    $r_{final} = r_{final} + (r_{t1} + r_{t2} + r_{t3})$
    if episode terminates $t = T$ **then**
      $r_t = r_{final}$
    else
      $r_t = r_t^{step}$
    end if
  end for
end for

constant current charging mode) and P_t^{DRL} is the ESS power determined by DRL Agent. U_t^E is the terminal voltage of ESS, and I_{ch}^E is the charging current (which is calculated from the charging rate r_{ch} and the ESS capacity B_E). The "$t-1$" notation indicates that the expert will gather the earlier system states (two time instants) when making the external intervention. In essence, the first condition of eqn. (22) helps to improve the training model quality in avoiding battery inactivity (which tends to happen without the external intervention), e.g., when the load is high; the second condition helps to encourage battery charging, e.g., when the renewable energy is abundant. Based on the intended design, charging of battery system is encouraged to take place when the electricity price is sufficiently low, and/or when the battery's SOC are very low in values. It is worth highlighting that, unlike the clear charging and discharging conditions adopted in optimization-based energy management, parts of the DRL's training logic, i.e., reward sub-component, may not follow the "deterministic" charging rule so long as the energy management's overarching goal is maintained through the full reward function design. MSRM algorithm is summarized in Algorithm 1.

V. Deep Reinforcement Learning Method

DRL fits a value function or a policy function using a deep neural network (DNN) and is capable of handling higher-dimensional state spaces than the regular RL. This means that, given sufficient data, DNN can be trained to replicate any desired input-output relationship. With reference to [15], [40], two common yet distinctive DRL algorithms are selected here as the control methods of microgrid distributed energy dispatching, namely double deep Q-network (DDQN) and policy gradient (PG). These are two of the DRL agents available in MATLAB's Reinforcement Learning Toolbox on which the proposed MSRM will be assessed. Basic definition of these two learning agents is summarized in what follows.

A. Double Deep Q-Network (DDQN)

Deep Q-network (DQN) technique is incorporated with the following features:

1) DQN uses a deep convolutional neural network to approximate the value function to handle a larger state space;
2) DQN uses replay memory to train the learning process to break the correlation between data;
3) DQN independently sets up a target critic network to calculate the action-value function of the temporal difference (TD) target to improve the stability during training.

A well acknowledged shortcoming of DQN is its inability to overcome the overestimation problem, in which the estimated value function is greater than the true value function. Overestimation will result in a suboptimal final policy. To address this issue, the DDQN method is proposed in [24]. DDQN implements action selection and evaluation using a variety of different value functions, as shown below [24]:

$$\theta_{t+1}{}' = \theta_t{}' + \alpha\left[R_t + \gamma Q\left(S_{t+1}, \arg\max_{\alpha_{t+1}} Q\left(S_{t+1}, a_{t+1}; \theta_t^-\right); \theta_t{}'\right) - Q\left(S_t, a_t; \theta_t{}'\right)\right]\nabla_{\theta_t{}'} Q\left(S_t, a_t; \theta_t{}'\right) \tag{26}$$

where action's choice is determined by the action-value function $Q(S_{t+1}, a_{t+1}; \theta_t^-)$, and the action's evaluation is determined by the action-value function $Q(S_{t+1}, \arg\max_{\alpha_{t+1}} Q(S_{t+1}, a_{t+1}; \theta_t^-); \theta_t{}')$.

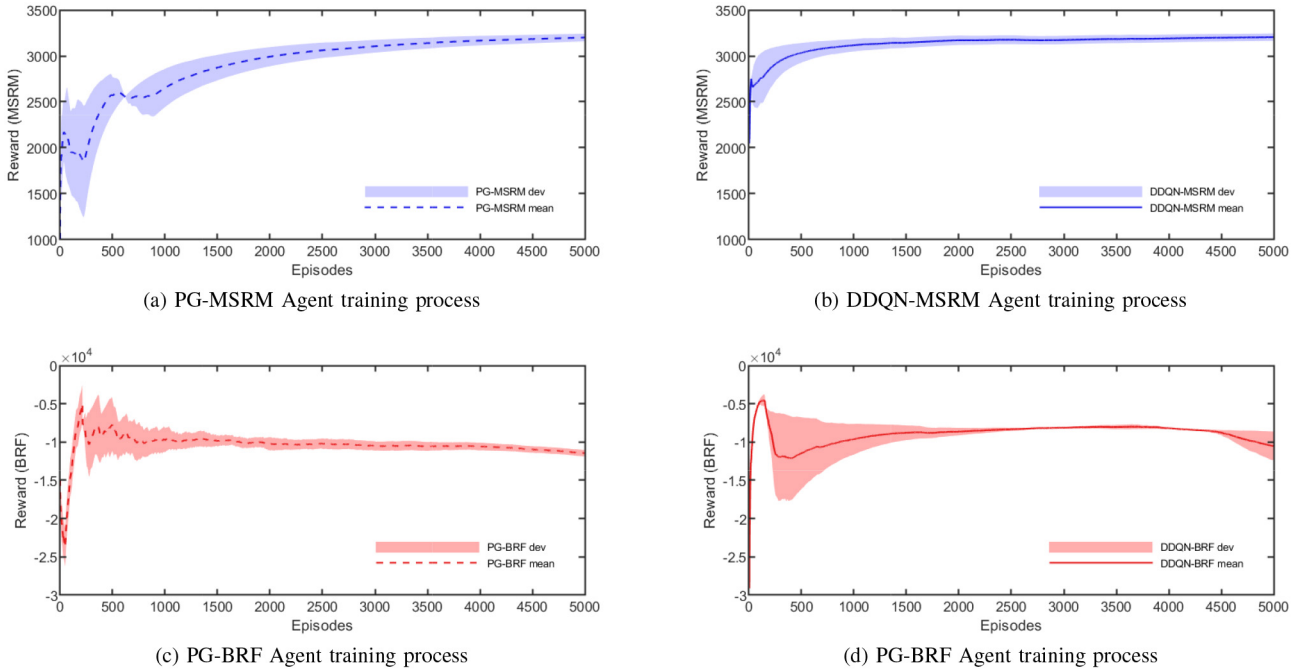
B. Policy Gradient

Policy gradient algorithm [25], [41] is a policy-based reinforcement learning method. The policy-based method is to use a parameterized linear function or a nonlinear function (such as DNN) to represent the policy as $\pi_\theta(a|s)$. Then, the optimal parameter θ that maximizes the expected cumulative return is to be found. The expected cumulative return is expressed as follow [41].

$$U(\theta) = \int_m P_\theta(\tau) R(\tau) d\tau \tag{27}$$

where m is the set of all possible trajectories, $P_\theta(\tau)$ is the probability of sequence τ occurring under the policy $\pi_\theta(a|s)$, and $R(\cdot)$ is the cumulative return. To maximize the expected cumulative return, we calculate the gradient for parameter θ, as shown [41]:

$$\begin{aligned}\nabla_\theta U(\theta) &= \int_m \nabla_\theta P_\theta(\tau) R(\tau) d\tau \\ &= \int_m P_\theta(\tau) \frac{\nabla_\theta P_\theta(\tau) R(\tau)}{P_\theta(\tau)} d\tau \\ &= \int_m P_\theta(\tau) \nabla_\theta \log P_\theta(\tau) R(\tau) d\tau \\ &= E\{\nabla_\theta \log P_\theta(\tau) R(\tau)\}\end{aligned} \tag{28}$$



(a) PG-MSRM Agent training process

(b) DDQN-MSRM Agent training process

(c) PG-BRF Agent training process

(d) PG-BRF Agent training process

Fig. 4. Training on the same data set that the line represents the mean of many trainings, the shadow region represent the deviation. Compare (a) and (c) to examine the impacts of MSRM and BRF in the PG algorithm, Compare (b) and (d) to examine the impacts of MSRM and BRF in the DDQN algorithm.

If the system is Markovian, $P_\theta(\tau)$ can be calculated from the joint probability of each variable in the sequence [25]:

$$P_\theta(\tau) = P(s_0)\prod_{t=0}^{T} P(s_{t+1}|s_t, a_t)\cdot \pi_\theta(a_t|s_t) \tag{29}$$

where $P(s_{t+1}|a_t, s_t)$ is the state transition probability, which has nothing to do with the strategy parameters and can be eliminated in the calculation of the strategy gradient [41]. Finally, the formula of the policy gradient can be expressed as [41]:

$$\nabla_\theta U(\theta) = E\left\{\sum_{t=0}^{T} \nabla_\theta \log\pi(a_t|s_t)R(\tau)\right\}. \tag{30}$$

VI. Simulation Results and Analysis

The low-voltage microgrid has been depicted in Fig. 1. It has an 800 kW wind farm, a 500 kW photovoltaic power plant, a 560kWh high-power-density energy storage system, and another 1190kWh high-capacity energy storage system. The load demand is configured in accordance with the Cardiff Council community's actual load profile [34]. The maximum and minimum limits of the battery's state of charge are set as 90% and 20%, respectively. Parameters of the distributed energy resources (DERs) and loads are tabulated in Table I. The load, solar power, and wind power profiles, as well as the UK real-time electricity price data, in 2018 are used. The agent's DNN utilizes the training data from January 27^{th} 2018 and October 16^{th} 2018, with 80% initial battery SOC. To ensure a fair comparison, the structure of the DNNs used in DDQN and PG are identical. DNN is comprised of three fully-connected layers: each layer has 100 neurons and it employs ReLU activation function. The MATLAB software is executed in a computing workstation with AMD Cores R5-5600X, 4.6GHz and 32 GB RAM. For a comprehensive benchmarking, this section analyzes four algorithms: DDQN-MSRM, PG-MSRM, DDQN-BRF, and PG-BRF with their convergence ability and energy management performance elaborated later. In Section VI-C, the DRL methods are comprehensively assessed with the model predictive control based energy scheduling methods [42], [43].

TABLE I
Parameter of Distributed Energy Resources and Loads

DG type	P^{DG}_{min}(kW)			P^{DG}_{max}(kW)		
Wind turbine	0			800		
Solar PV	0			500		
Load type	P^{L}_{min}(kW)			P^{L}_{max}(kW)		
Critical load	0			580		
Noncritical load	0			620		
Energy storage	B_E	P^{ch}_{min}	P^{disch}_{max}	SOC_{min}	SOC_{max}	C-rate
HCE	1190	595	595	20	90	0.5
HPE	560	840	840	20	90	1.5

Broadly speaking, during the initial stages of training for each algorithm, the agent will make numerous random choices. This causes the agent's reward to fluctuate significantly. Despite earlier fluctuation, the overall reward is on the upward trend. As the training covers more episodes, the reward progressively stabilizes and the training converges, which indicates that the agent can now choose the action strategy that is more consistent and closer to the application's objective.

A. Convergence Capabilities

This subsection evaluates the performance of MSRM in improving the convergence characteristic over the baseline reward function. All the algorithms were trained using the same data set with the same samples order. Fig. 4 depicts

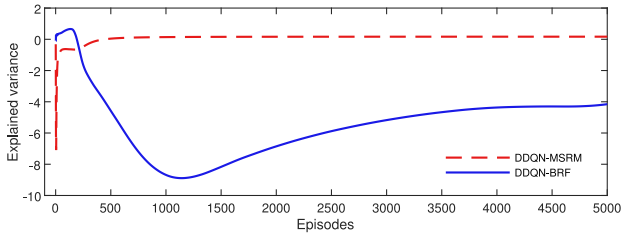


Fig. 5. Explained Varience between the true Q value and the estimated Q value under the same test state set.

the training procedure for all four algorithms. In the early exploration stage, the BRF's reward varies significantly. After approx. 1000 episodes, the reward begins to converge but with a downward trend (especially for DDQN-BRF). The results indicate that the BRF-based DRL algorithm tends converge to a suboptimal solution after a somewhat significant volatility during the exploration stage, take longer time to improve in terms of reward outcome. In comparison, the MSRM-based DRL algorithm continues to enhance reward expectations after the discovery stage in a gradual manner and stabilizes after about 2000^{th} episodes (DDQN-MSRM) or 3500^{th} episodes (PG-MSRM). Both results hint about the ability of MSRM in providing more gradual but accurate guidance and in ensuring a more reliable learning outcome.

What follows analyzes the performance of the reward mechanism/function in conjunction with DDQN algorithm. The basis of analysis is that since the Q-value updates of the Q-learning-based algorithms are calculated based on the estimated Q-values, the discrepancy in the estimated value (from the true value) may lead to over-actions and eventually leading to suboptimal training outcome. Explained variance can be calculated as follows [44]:

$$explained\ variance = 1 - \frac{var(value^{ture} - value^{estimate})}{var(value^{ture})} \quad (31)$$

where the range is $(-\infty, 1]$. "1" means that the estimated and true values are essentially the same. Fig. 5 describes the Explained Variance between the true Q value and the estimated Q value of the DDQN-MSRM and DDQN-BRF, respectively. It is evidenced that MSRM is consistent in resulting in small variance whereas BRF tends to result in overestimation of the value function.

B. Energy Management Performance

The trained agents from the four DRL algorithms (as described in Section VI-A) are subject to further assessment of their energy management performance. Fig. 6 depicts the typical wind turbine, photovoltaic, and load day-profiles together with the real-time electricity price. The ESS and grid power scheduling and the corresponding ESS's SOCs are summarized in Figs. 7 and 8, and the time interval Δt for scheduling is 30 minutes. Fig. 7 shows that, as compared to RBF-agent, MSRM-agent results in a better tracking of the renewable energy output by better utilizing the two ESS systems. The ESS is charged during the periods of high renewable energy input and is discharged during the periods of peak loadings. On

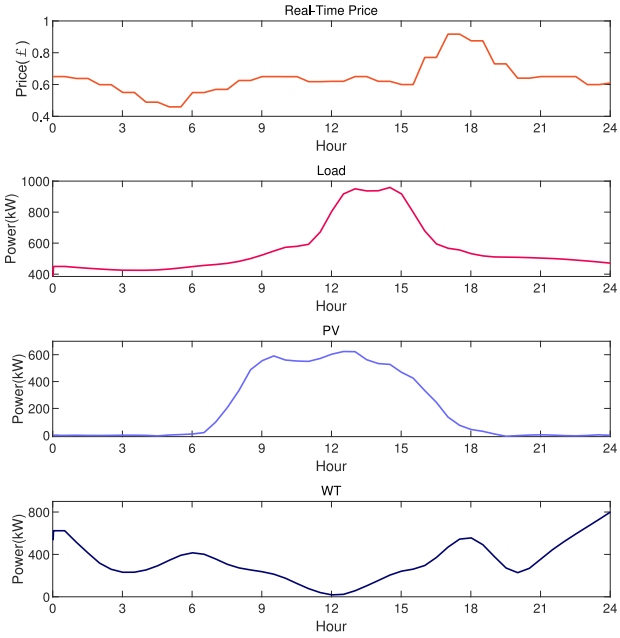


Fig. 6. Profiles of PV, WT, LOAD and electricity prices in a typical day.

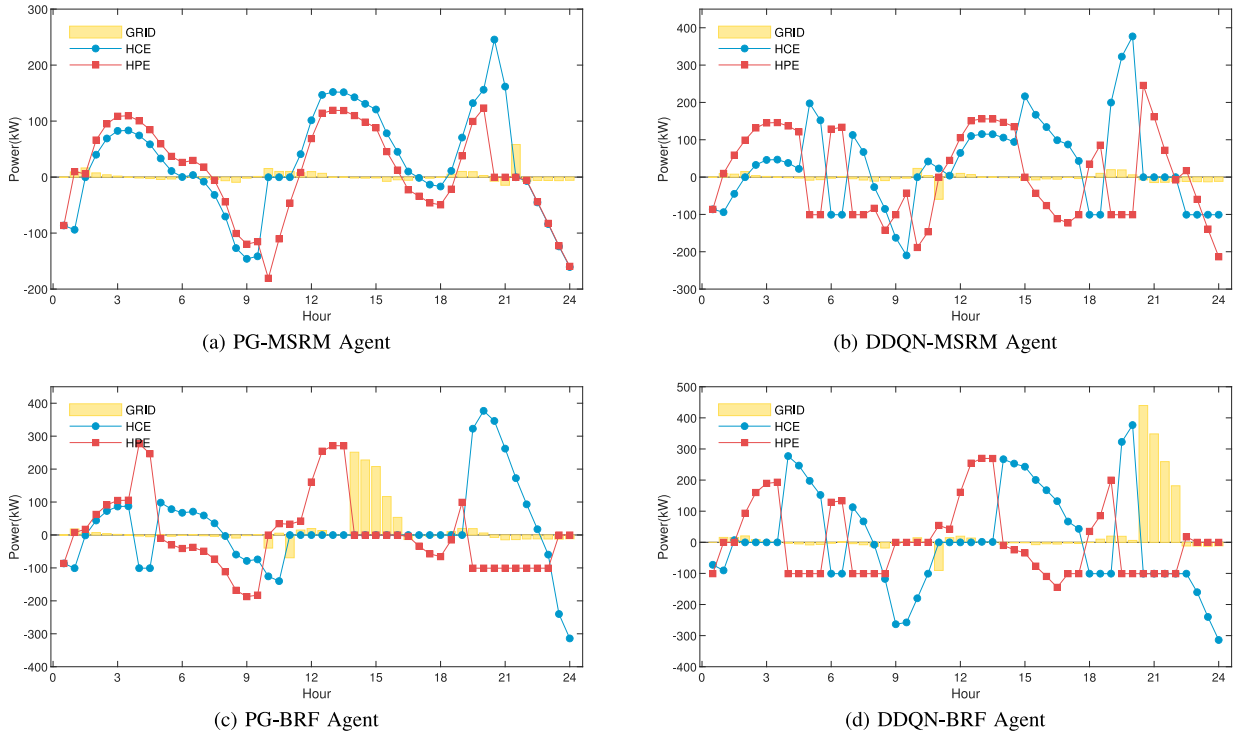
TABLE II
THE TIME TO TRIGGER THE BATTERY PROTECTION MECHANISM IN A TYPICAL DAY

	PG-MSRM	DDQN-MSRM	PG-BRF	DQN-BRF
HPE over charge (min)	0	0	33	133
HPE over discharge (min)	10	4	76	18
HCE over charge (min)	0	0	470	173
HCE over discharge (min)	20	15	0	0

the other hand, the RBF-agent consumes more electricity from main grid and under-utilize the available renewable energy resources. It is worth highlighting that the focus here is on the likelihood of the DRL agents to be trained into converging to an improved energy management performance. This is analogous to improving the accuracy of the stochastic short-term forecast models in conventional, dynamic-programming-based EMS algorithms.

Fig. 8 shows the SOC values of the ESSs for all four algorithms. Figs. 8c and 8d show that the BRF-based agents tend to activate the overcharge and over-discharge protection mechanisms, as compared to MSRM-based agents. Table II summarizes the duration of triggered battery protections. It clearly shows that the BRF-based agents have longer inoperable period as compared MSRM-based agents, especially for the overcharge scenario. The over-discharging situation in Table II can be explained primarily by the pursuit of the behavior of DDQN-MSRM and PG-MSRM towards discharging due to the self-balancing requirement (based on eqn. (19)), leading to slight over-discharging. However, comparatively, the discharge duration is much smaller than those in BRF counterpart. Note also that the over-charging has been effectively mitigated by MSRM-based techniques.

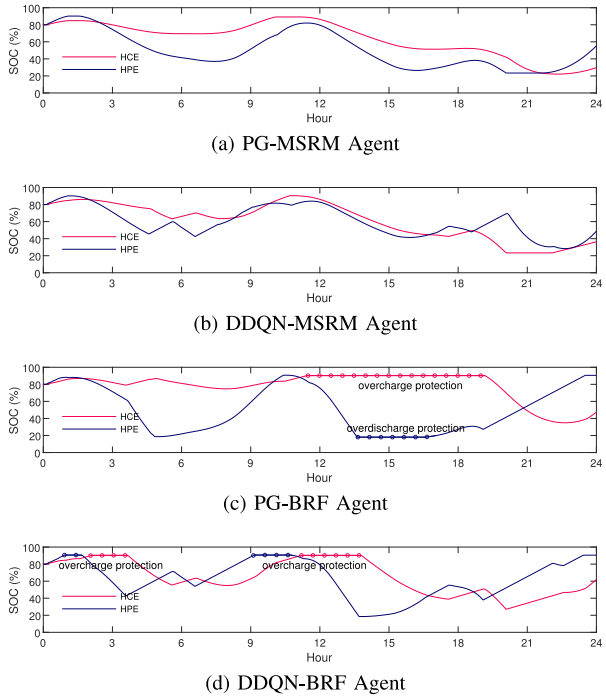
Lastly, Table III shows the operational results of the four agents for a week period. The results clearly hint that the



(a) PG-MSRM Agent (b) DDQN-MSRM Agent

(c) PG-BRF Agent (d) DDQN-BRF Agent

Fig. 7. Compare the scheduling results of different methods in a typical day.



(a) PG-MSRM Agent

(b) DDQN-MSRM Agent

(c) PG-BRF Agent

(d) DDQN-BRF Agent

Fig. 8. Battery SOC change in a typical day.

MSRM-based agent has the potential to produce a better energy management performance in terms of economic and technical performance (and consequently, the environmental performance too, e.g., CO_2 reduction, calculated using (9)). Incidentally, as compared to other algorithms, the PG-MSRM algorithm can minimize the carbon emissions of the distribution grid to as much as 30 tons per week and improve the self-balancing rate to 99.53%.

TABLE III
COMPARISON BETWEEN USING MSRM AND USING BRF IN A TYPICAL WEEK

	PG-MSRM	DDQN-MSRM	PG-BRF	DQN-BRF
Average daily cost (£)	43.6	92.1	214.14	268.14
self-balancing rate	99.53%	98.17%	96.79%	95.56%
Meet reliability	YES	YES	YES	YES
CO_2 reductions (ton)	30.57	29.87	28.02	27.93

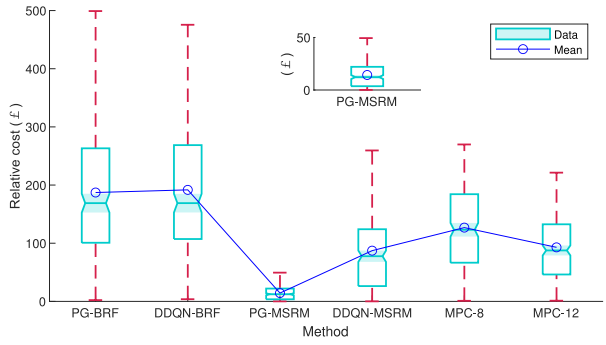
*YES means that the reliability rate is above 0.95, which meets the reliability requirements.

C. Comparison With MPC-Based Scheduling

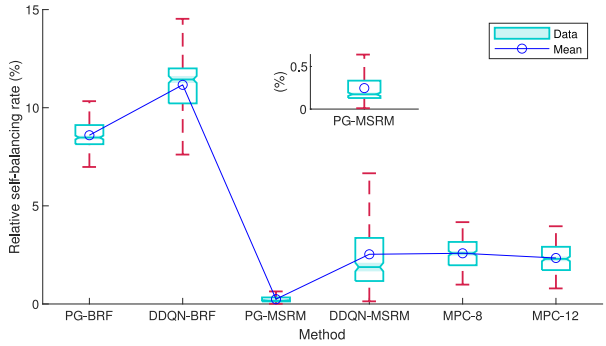
The four methods are further compared and analyzed with two MPC-based scheduling methods [42], [43]: MPC-8 and MPC-12. They are rolling-horizon optimization with control horizon of 8 and 12 steps, and with real-time forecast [42], [43]. The real-time forecast is simplified here through the addition of random error (max 10%), imitating forecast error, to the original data [17]. The time interval Δt of each step is 30 minutes. In addition, all methods are benchmarked against the optimal scheduling which assumes that all future forecast is accurately known. The objective function used by the above methods is show in eqn. (32) and is constrained by eqns. (2)-(5) [42], [43], [45]. The power balance is considered as a hard equality constraint, as in eqn. (33).

$$minimize \quad \left(Cost_t^{Grid} + D_t^{self-balan}\right) \tag{32}$$

$$P_t^{PV} + P_t^{WT} + P_t^{HPE} + P_t^{HCE} + P_t^{G} = P_t^{L} \tag{33}$$



(a) Comparison of costs



(b) Comparison of self-balancing rate

Fig. 9. The performance of different methods is compared through 250 test scenarios in box plot; use a solid blue line with a circle mark to indicate the mean of each group of tests.

All methods will be run in 250 test scenarios. The test scene is different from the scene in the Section VI-A training set. The ideal optimal scheduling serves as the benchmark to calculate relative scores, which measure the gaps between the methods and the best schedule. The relative cost is calculated through:

$$Cost_{rel}^{Grid} = \left| Cost_{optimal}^{Grid} - Cost_{other}^{Grid} \right| \tag{34}$$

where $Cost_{optimal}^{Grid}$ is the operating cost calculated using the optimal scheduling, and $Cost_{other}^{Grid}$ is corresponding counterpart for the DRL methods and MPC-8/12. The relative self-balancing rate $R_{rel}^{self-balan}$ is calculated using:

$$R_{rel}^{self-balan} = \left| R_{optimal}^{self-balan} - R_{other}^{self-balan} \right| \tag{35}$$

where $R_{optimal}^{self-balan}$ is the self-balancing rate calculated using the optimal scheduling, and $R_{other}^{self-balan}$ is the corresponding counterpart for the DRL methods and MPC-8/12. Since power balance is assumed, reliability rate is not applicable to MPC-8/12.

The comparison results are summarized in the form of box plots in Fig. 9. Firstly, PG methods generally perform better than DDQN methods. This somewhat agrees with the expectation of PG deals better with large action space as compared to DQN methods [26]. Secondly, the difference between the PG-MSRM and the optimal strategy is small, having the average relative cost of £14.3 and the average relative self-balancing rate of 0.245%. This represents a significant improvement as compared to PG-BRF and DDQN methods. Thirdly, MPC methods generally perform better than DRL methods based on BRF, with MPC-12 (average relative cost is £92.91; average relative self-balancing rate is 2.34%) being slightly better than MPC-8 due to longer control horizon.

TABLE IV
COMPARISON OF ONLINE EXECUTION TIME IN ONE SCENARIO (48 STEPS) OF DIFFERENT METHODS (UNIT:S)

	PG-MSRM	DDQN-MSRM	PG-BRF	DQN-BRF	MPC-8	MPC-12
Time(s)	1.23	1.31	1.21	1.19	28.62	76.45

The online execution time of various methods is tabulated in Table IV. It is seen that DRL methods require a fraction of time as compared to MPC methods. This is expected as MPC requires real-time optimization while DRL agents has much simpler computation with the trained neural network. The advantage of DRL methods will likely be more obvious in more-complex system (e.g., those require AC power flow model [8]) and system with smaller time step (e.g., the time interval Δt of each step is 15 minutes in [42]). Finally, the implementation difficulties of these methods can be briefly summarized as follows: both DRL and MPC method require a large amount of environmental data to train/build the agent/model. While MPC faces with the problem of accurate forecasting model and prediction accuracy, DRL also faces the problems of long offline training and evaluating suitability and effectiveness of reward function design.

VII. CONCLUSION

This paper proposes and investigates a multi-stage reward mechanism aiming to improve the sparse reward problem encountered in the deep reinforcement learning based microgrid energy management. As compared to the ones with baseline reward function, the MSRM-based DRL algorithms demonstrated consistent performance in improving the training quality in terms of convergence, explained variance (DDQN only), energy management performance, as well as battery over-charging protection. The simulation has been conducted using the actual load data from the Cardiff Council community. The methods have also benchmarked with the optimal and MPC-based scheduling strategies, and it is shown that PG-MSRM performs well in terms of relative cost, self-balancing rate, and computational time. It is hoped that the proposed assessment can inform the deep reinforced learning-based energy management's state of the arts and contribute to promoting the adoption in naturally stochastic microgrid infrastructure.

REFERENCES

[1] Z. Wang, W. Wu, and B. Zhang, "A fully distributed power dispatch method for fast frequency recovery and minimal generation cost in autonomous microgrids," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 19–31, Jan. 2016.

[2] R. H. Lasseter and P. Paigi, "Microgrid: A conceptual solution," in *Proc. IEEE 35th Annu. Power Electron. Spec. Conf.*, vol. 6, Jun. 2004, pp. 4285–4290.

[3] R. Lasseter, "MicroGrids," in *Proc. IEEE Power Eng. Soc. Winter Meeting. Conf.*, vol. 1, Jan. 2002, pp. 305–308.

[4] H. Kanchev, F. Colas, V. Lazarov, and B. Francois, "Emission reduction and economical optimization of an urban microgrid operation including dispatched PV-based active generators," *IEEE Trans. Sustain. Energy*, vol. 5, no. 4, pp. 1397–1405, Oct. 2014.

[5] K. Rahbar, J. Xu, and R. Zhang, "Real-time energy storage management for renewable integration in microgrid: An off-line optimization approach," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 124–134, Jan. 2015.

[6] T. Wang, H. Kamath, and S. Willard, "Control and optimization of grid-tied photovoltaic storage systems using model predictive control," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 1010–1017, Mar. 2014.

[7] W. Shi, N. Li, C.-C. Chu, and R. Gadh, "Real-time energy management in microgrids," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 228–238, Jan. 2017.

[8] P. Zeng, H. Li, H. He, and S. Li, "Dynamic energy management of a microgrid using approximate dynamic programming and deep recurrent neural network learning," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 4435–4445, Jul. 2019.

[9] Y. Du and F. Li, "Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1066–1076, Mar. 2020.

[10] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[11] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, p. 1054, Sep. 1998.

[12] Y. Li, "Deep reinforcement learning: An overview," 2018, *arXiv:1701.07274*.

[13] E. O. Arwa and K. A. Folly, "Reinforcement learning techniques for optimal power control in grid-connected microgrids: A comprehensive review," *IEEE Access*, vol. 8, pp. 208992–209007, 2020.

[14] X. Qiu, T. A. Nguyen, and M. L. Crow, "Heterogeneous energy storage optimization for microgrids," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1453–1461, May 2016.

[15] E. Mocanu *et al.*, "On-line building energy optimization using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3698–3708, Jul. 2019.

[16] Z. Wen, D. C. O'Neill, and H. R. Maei, "Optimal demand response using device-based reinforcement learning," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2312–2324, Sep. 2015.

[17] H. Li, Z. Wan, and H. He, "Constrained EV charging scheduling based on safe deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2427–2439, May 2020.

[18] X. Lu, X. Xiao, L. Xiao, C. Dai, M. Peng, and H. V. Poor, "Reinforcement learning-based microgrid energy trading with a reduced power plant schedule," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10728–10737, Dec. 2019.

[19] R. Lu, S. H. Hong, and M. Yu, "Demand response for home energy management using reinforcement learning and artificial neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6629–6639, Nov. 2019.

[20] X. Xu, Y. Jia, Y. Xu, Z. Xu, S. Chai, and C. S. Lai, "A multi-agent reinforcement learning-based data-driven method for home energy management," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3201–3211, Jul. 2020.

[21] G. K. Venayagamoorthy, R. K. Sharma, P. K. Gautam, and A. Ahmadi, "Dynamic energy management system for a smart microgrid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1643–1656, Aug. 2016.

[22] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Sep. 2019.

[23] H. Song, Y. Liu, J. Zhao, J. Liu, and G. Wu, "Prioritized replay dueling DDQN based grid-edge control of community energy storage system," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 4950–4961, Nov. 2021.

[24] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," 2016, *arXiv:1509.06461*.

[25] R. S. Sutton, D. A. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. NIPS*, 1999, pp. 1057–1063.

[26] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2016, *arXiv:1509.02971*.

[27] N. Heess *et al.*, "Emergence of locomotion behaviours in rich environments," 2017, *arXiv:1707.02286*.

[28] L. Tightiz and H. Yang, "Resilience microgrid as power system integrity protection scheme element with reinforcement learning based management," *IEEE Access*, vol. 9, pp. 83963–83975, 2021.

[29] L. Lei, Y. Tan, G. Dahlenburg, W. Xiang, and K. Zheng, "Dynamic energy dispatch based on deep reinforcement learning in IoT-driven smart isolated microgrids," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7938–7953, May 2021.

[30] B. Huang and J. Wang, "Deep-reinforcement-learning-based capacity scheduling for PV-battery storage system," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2272–2283, May 2021.

[31] Z. Qin, D. Liu, H. Hua, and J. Cao, "Privacy preserving load control of residential microgrid via deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 12, no. 5, pp. 4079–4089, Sep. 2021.

[32] T. Degris, M. White, and R. S. Sutton, "Off-policy actor-critic," 2012, *arXiv:1205.4839*.

[33] S. Ivanov and A. D'yakonov, "Modern deep reinforcement learning algorithms," 2019, *arXiv:1906.10025*.

[34] "Cardiff Council Carbon Culture." 2021. [Online]. Available: https://platform.carbonculture.net/communities/cardiff-council/19/

[35] Z. Quanming, Z. Xiaodi, S. Ke, Z. Dan, and T. Wei, "A grid-connected microgrid optimal allocation method considering self-balancing rate," *J. Phys. Conf. Ser.*, vol. 1659, no. 1, 2020, Art. no. 12023.

[36] "Greenhouse Gas Reporting: Conversion Factors 2018." 2018. [Online]. Available: https://www.gov.uk/government/publications/greenhouse-gas-reporting-conversion-factors-2018

[37] R. Leo, R. S. Milton, and S. Sibi, "Reinforcement learning for optimal energy management of a solar microgrid," in *Proc. IEEE Global Humanitarian Technol. Conf. South Asia Satell. (GHTC-SAS)*, 2014, pp. 183–188.

[38] F. Ming, F. Gao, K. Liu, J. Wu, Z. Xu, and W. Li, "Constrained double deep Q-learning network for EVs charging scheduling with renewable energy," in *Proc. IEEE 16th Int. Conf. Autom. Sci. Eng. (CASE)*, 2020, pp. 636–641.

[39] Y. Liu, D. Zhang, and H. B. Gooi, "Optimization strategy based on deep reinforcement learning for home energy management," *CSEE J. Power Energy Syst.*, vol. 6, no. 3, pp. 572–582, Sep. 2020.

[40] V.-H. Bui, A. Hussain, and H.-M. Kim, "Double deep *Q*-learning-based distributed operation of battery energy storage system considering uncertainties," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 457–469, Jan. 2020.

[41] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Netw. Off. J. Int. Neural Netw. Soc.*, vol. 21, no. 4, pp. 682–697, 2008.

[42] A. Parisio, E. Rikos, and L. Glielmo, "A model predictive control approach to microgrid operation optimization," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 5, pp. 1813–1827, Sep. 2014.

[43] R. Palma-Behnke *et al.*, "A microgrid energy management system based on the rolling horizon strategy," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 996–1006, Jun. 2013.

[44] J. A. Camacho, A. K. Smilde, E. Saccenti, and J. A. Westerhuis, "All sparse PCA models are wrong, but some are useful. Part I: Computation of scores, residuals and explained variance," 2019, *arXiv:1907.03989*.

[45] Z. Liu, S. Liu, Q. Li, Y. Zhang, W. Deng, and L. Zhou, "Optimal day-ahead scheduling of islanded microgrid considering risk-based reserve decision," *J. Modern Power Syst. Clean Energy*, vol. 9, no. 5, pp. 1149–1160, Sep. 2021.

Hui Hwang Goh (Senior Member, IEEE) received the B.Eng. (Hons.) and M.Eng. degrees in electrical engineering and the Ph.D. degree in electrical engineering from Universiti Teknologi Malaysia, Johor Bahru, Malaysia, in 1998, 2002, and 2007, respectively.

He is currently a Professor of Electrical Engineering with the School of Electrical Engineering, Guangxi University, Nanning, China. His research interests include embedded power generation modeling and simulation, power quality studies, wavelet analysis, multicriteria decision-making, renewable energies, and dynamic equivalent. He is also a Fellow of the Institution of Engineering and Technology, U.K., the ASEAN Academy of Engineering and Technology, and The Institution of Engineers, Malaysia, a Chartered Engineer under the Engineering Council United Kingdom, and a Professional Engineer under the Board of Engineers, Malaysia. He is also the Foreign Fellow of Chinese Society for Electrical Engineering.

Yifeng Huang was born in Guangdong, China, in 1997. He received the B.Eng. degree in electrical engineering from Guangdong Polytechnic Normal University, Guangdong, China, in 2019. He is currently pursuing the M.S. degree with the College of Electrical Engineering, Guangxi University, Guangxi, China. His current research interests include microgrid, deep reinforcement learning, and energy management system.

Chee Shen Lim (Senior Member, IEEE) received the B.Eng. degree (Hons.) in electrical engineering from the University of Malaya, Malaysia, in 2009, and the joint-university Ph.D. degrees in power electronics and drives from the University of Malaya and Liverpool John Moores University, U.K., in 2013.

He was a Research Scientist with the Experimental Power Grid Center, A*STAR, Singapore, from 2013 to 2015. He has been with the University of Southampton Malaysia since November 2015. He is currently an Associate Professor of Electrical and Electronic Engineering with Xi'an Jiaotong-Liverpool University. His research interests include advanced model predictive control design, multiphase motor drives, grid-connected converter control, and microgrid hierarchical control. He also serves as an Associate Editor for the *IET Electric Power Applications*.

Dongdong Zhang (Member, IEEE) was born in Jining, China, in 1990. He received the B.Eng. degree in electrical engineering and automation from the College of Electrical Engineering, Qilu University of Technology, Shandong, China, in 2013, the M.S. degree in electric power system and automation from North China Electric Power University in 2016, and the Ph.D. degree in electrical engineering from Xi'an Jiaotong University in 2019.

He is currently an Assistant Professor with Guangxi University. His research interests include the modeling and optimization of multiple energy system, energy market, and electrical machines and its driving system design.

Hui Liu (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from the College of Electrical Engineering, Guangxi University, China, in 2004 and 2007, respectively. He worked with Tsinghua University as a Postdoctoral Fellow from 2011 to 2013 and in Jiangsu University as a Faculty Member from 2007 to 2016. He visited the Energy Systems Division with Argonne National Laboratory, Argonne, IL, USA, from 2014 to 2015. He joined the School of Electrical Engineering WITH Guangxi University in 2016, where he is a Professor and the Deputy Dean. He is an Editor of the IEEE TRANSACTIONS ON SMART GRID and the IEEE PES LETTERS. He is also an Associate Editor of the *IET Smart Grid* and the *IET Generation, Transmission & Distribution*. His research interests include power system optimization, power system stability and control, electric vehicles, integrated energy systems, and demand response.

Wei Dai (Member, IEEE) received the Ph.D. degree in electrical engineering from Chongqing University, in 2018. He currently works as an Assistant Professor with Guangxi University. His research interests include multiple energy systems, power systems analysis, renewable energy, and large-scale system problems.

Tonni Agustiono Kurniawan received the Ph.D. degree from the Hong Kong Polytechnic University. He is an Associate Professor with Xiamen University, China. Prior to joining the University, he was a Scholar with United Nations University, Tokyo. Significant contribution to research in the field has earned him recognition from the World Economic Forum, Switzerland. So far he has been cited over 7,600 times with a Hirsch factor of 28 (Scopus). Since 2011, the Institute for Scientific Information-Thompson Reuters has identified him among the top 1% of researchers in the field of engineering according to the Essential Science Indicators of Web of Knowledge.

Saifur Rahman (Life Fellow, IEEE) received the B.Sc. degree in electrical engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 1973, and the Ph.D. degree in electrical engineering from Virginia Polytechnic Institute and State University, Arlington, VA, USA, in 1978.

He is the Director of the Advanced Research Institute, Virginia Polytechnic Institute and State University, where he is the Joseph Loring Professor of Electrical and Computer Engineering and also directs the Center for Energy and the Global Environment. He has authored or coauthored in the areas of his research interests which include smart grid, conventional and renewable energy systems, load forecasting, uncertainty evaluation, infrastructure planning, and IoT device integration. He served as the President of IEEE Power and Energy Society for 2018 and 2019. He was the Vice President of the IEEE Publications Board and as a member of the IEEE Board of Directors in 2006. He was the Founding Editor-in-Chief of the IEEE TRANSACTIONS ON SUSTAINABLE ENERGY and *IEEE Electrification Magazine*. He is a Distinguished Lecturer of the IEEE Power and Energy Society. From 1996 to 1999, he was a Program Director with the Engineering Directorate of the National Science Foundation. He was elected President-elect of IEEE in 2021.