Resilient Distribution Networks by Microgrid Formation Using Deep Reinforcement Learning

Yuxiong Huang, *Student Member, IEEE*, Gengfeng Li, *Member, IEEE*, Chen Chen, *Senior Member, IEEE*, Yiheng Bian, *Graduate Student Member, IEEE*, Tao Qian, *Student Member, IEEE*, and Zhaohong Bie, *Senior Member, IEEE*

***Abstract*—Resilience becomes vital for power grids facing the increasingly frequent extreme weather events. Microgrid formation is a promising way to achieve resilient distribution networks (RDN) when the utility power is unavailable. This paper proposes a RDN-oriented microgrid formation (RoMF) method based on the deep reinforcement learning (DRL) technique, which integrates the OpenDSS as an interaction object and searches for optimal control policies in a model-free fashion. Specifically, we formulate the microgrid formation problem as a Markov decision process, taking into account complex factors such as unbalanced three-phase power flow and microgrid operation constraints. Next, a simulator-based RoMF environment is constructed and integrated into the OpenAI Gym, providing a standard agent-environment interface for applying DRL algorithms. Then, the deep Q-network is used to search for optimal microgrid formation strategies, and an offline-training and online-application framework of the DRL-based RoMF is given. Finally, extensive numerical results validate the effectiveness of our proposed method.**

***Index Terms*—Critical load, deep learning, distributed generator, distribution network, load restoration, microgrid, reinforcement learning, resilience.**

I. Introduction

THE RELIABLE and efficient operation of power systems is essential for the new global economy. Grid resilience refers to the capability to endure and curtail the adverse impact of disruptions, comprising three main factors, i.e., absorb, adapt, and recover from extreme events [1]. For the power distribution network, recent breakthrough smart grid technologies enable it to enhance its resilience via the microgrid formation when the utility power from main grids is lost. The microgrid formation refers to using distributed generators (DGs) to supply loads at distribution feeders and form multi islanded microgrids with dynamic boundaries [2]. By adopting such a bottom-up load restoration strategy in parallel with conventional top-down strategies, i.e., the reenergization of transmission networks using bulk generators, grid resilience can be considerably improved.

The resilient distribution network (RDN) has become a essential research field for the planning and operation of power systems. The factors influencing grid resilience were explored in [3], [4]. A generic research framework for assessing the impact of weather on grid resilience was outlined in [5]. The modeling approaches for energy infrastructure resilience were investigated in [6]. A review on resilience studies in active distribution systems was given in [7]. The hardening and operational measures toward distribution system resilience were discussed in [8].

For low-probability or extreme events, the restoration measures in the post-disruption stage are considered vital. As per IEEE Std. 1547-2018 [9], microgrids can improve grid resilience by providing the islanded power during an area outage. Moreover, microgrid formation as a potential solution for resilience enhancement has become a major focus of research [10]. A comprehensive review on modeling and operational strategies on microgrids and resilience is given in [11]. A critical load restoration (CLR) approach by forming multiple microgrids energized by DGs was proposed in [2], and a self-healing strategy for the distribution system with both dispatchable and nondispatchable DGs was proposed in [12]. Subsequently, more factors, e.g., topology reconfiguration [13], [14], multi-energy microgrids [15], mobile energy storage [16], demand response [17], renewable energy resources [18], [19], damage assessment [20], the adequacy of generation resources [21], [22], frequency dynamics [23], microgrid stability and dynamic performance of DGs [24] have been explored in microgrid-based CLR studies.

Notably, most existing CLR studies are carried out based on Mathematical Programming (MP), where the CLR problem is formulated as a mixed-integer (or integer) linear (or nonlinear) program and solved with optimization algorithm-based solvers. For MP-based studies, building an explicit mathematical model that functions as a digital twin of the research object while meeting the requirements of optimization algorithms for solvability is the basis of their optimality and robustness. Such model-based schemes require much effort in establishing specific models and thus has a high threshold for practical application, and may suffer in complex object modeling, new feature embedding, and computational efficiency for large-scale systems, that deserve our attention.

Manuscript received 9 September 2021; revised 8 January 2022 and 9 April 2022; accepted 24 May 2022. Date of publication 1 June 2022; date of current version 21 October 2022. This work was supported by the National Natural Science Foundation of China under Grant 51977168. Paper no. TSG-01450-2021. *(Corresponding author: Gengfeng Li.)*

The authors are with the School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: hyx.xj@stu.xjtu.edu.cn; gengfengli@mail.xjtu.edu.cn; morningchen@xjtu.edu.cn; byh0xyz@stu.xjtu.edu.cn; taylorqian@stu.xjtu.edu.cn; zhbie@mail.xjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2022.3179593.

Digital Object Identifier 10.1109/TSG.2022.3179593

Machine Learning (ML) provides a promising way to tackle these challenges.[1] Specifically, reinforcement learning (RL) has recently been applied to address many power system control tasks such as voltage control [25], [26], demand response [27], [28], [29], electric vehicle charging navigation [30], [31], network reconfiguration [32], power management of networked microgrid [33], [34], and resilience operation of distribution networks [35]. This paper studies the RDN-oriented microgrid formation (RoMF) using a model-free[2] deep reinforcement learning (DRL) scheme, and optimal control policies are directly learned through the simulator-based agent-environment interaction (SAEI) [36]. Compared to the MP-based environment in [35], this paper implements a simulator-based environment, where the OpenDSS is integrated to perform unbalanced three-phase power flow calculations. Since simulators are generally feature-rich and not limited to a specific task, learning from SAEI can facilitate complex object-oriented learning and new feature embedding, and thus seems adaptive for real-world applications. In addition, the knowledge learned by DRL is represented as neural network parameters and applied based on input-output mapping, which supports a near-zero time-consuming decision-making style.

Specifically, the RoMF is used to enhance the distribution network resilience through restoring critical loads at distribution feeders with DGs when the utility power from main grids is lost. Moreover, the RoMF is a dynamic problem, indicating that the decision-making of the microgrid formation is performed sequentially, and the decision at each time step takes into account current observable states and future uncertainties. To achieve that, the RoMF problem is first analysed, including the object and constraints. Next, the RoMF problem is formulated into a Markov decision process (MDP), and a RoMF simulation environment is constructed based on the OpenDSS. The implemented environment is integrated into the OpenAI Gym [37], providing a standard agent-environment interface to perform DRL algorithms. Then, we apply the deep Q-network [38], [39] to search for optimal control policies, and an offline-training and online-application framework of the DRL-based RoMF scheme is proposed. At last, case studies based on three modified IEEE test feeders were conducted, where other learning-based methods and the MP method in [2] are used for comparison to validate the proposed method.

Main contributions of this paper include: 1) By formulating the RoMF problem as a MDP, and establishing a simulator-based environment where the OpenDSS is integrated to perform unbalanced three-phase power flow calculations, it provides a new model-free scheme for the microgrid formation problem with complex scenarios. 2) By introducing the DRL framework, and implementing an offline-training and online-application mode, a microgrid formation agent is built in this paper, which can provide near-optimal solutions with extremely fast application efficiency.

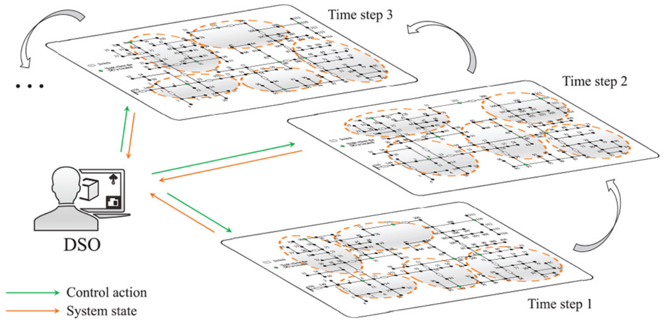


Fig. 1. The architecture of RDN-oriented microgrid formation.

The remainder of this paper is organized as follows: Section II is the problem formulation, including the introduction of the RoMF problem and its MDP formulation, and the implementation of the simulator-based environment. Section III introduces the DRL algorithm used to search for optimal policies, and the training-application framework of the DRL-based RoMF. Case studies are presented in Section IV, followed by Section V that concludes this paper.

II. Problem Formulation

In this section, the RoMF problem is first introduced, and its MDP formulation is given by defining corresponding MDP elements (i.e., action, state, reward, etc.). Then, the implementation of the SAEI is presented.

A. RDN-Oriented Microgrid Formation

The distribution network is denoted as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ and $\mathcal{E}$ denote the set of nodes and the set of edges (i.e., power lines), respectively, $|\mathcal{N}| = N$, $|\mathcal{E}| = E$. The set of nodes that DGs connected to is denoted by $\mathcal{M} \subset \mathcal{N}$, $|\mathcal{M}| = M$. The set of sectionalizing switches is denoted by $\mathcal{W}$, $|\mathcal{W}| = W$. The real and reactive power demands at each node $i \in \mathcal{N}$ are represented as $p_{t,i,\phi}$ and $q_{t,i,\phi}$, respectively, where subscript t denotes the time step, and ϕ denotes the phase. The weight coefficient associated with the load at node i is denoted by ω_i.

A major fault, occurred at the main grid due to natural disasters or other extreme events, may result in that the distribution network cannot be energized through the main grid for a period of time. During the outage, by means of switch devices and DGs, the distribution system operator (DSO) can disassemble the distribution network into multiple islanded microgrids to sustean power services. For the RoMF problem, as shown in Fig. 1, the post-disruption recovery period is discretized and denoted as a set $\mathcal{T}$, $|\mathcal{T}| = T$. At each time step $t \in \mathcal{T}$, the DSO will generate and execute a microgrid formation strategy. Specifically, the strategy denotes a combination of switch operations that is used to determine the supply areas of islanded microgrids. Its aim is to maximize the expected cumulative restored load throughout the recovery period while satisfying operational constraints. Such a dynamic decision-making manner helps the DSO to generate a strategy that is optimal

[1] Both MP and ML are highly sophisticated advanced techniques, so these challenges are not inevitably applicable to all MP methods and solvable by any ML methods.

[2] The term "model-free" denotes that the agent learns without knowing the exact environment model, but a model-based RDN environment is still need for interacting with the agent.

for the whole post-disruption recovery while just observing current partial system states.

Before formulating the RoMF problem into a MDP, the object and operational constraints of the RoMF problem will be first discussed, which serve as the basics for defining corresponding MDP elements and modeling the SAEI.

1) Object of RoMF: At each time step, the DSO aims to maximize the expected sum of restored critical loads during whole restoration period. The object is formulated as:

$$\max\ \mathbb{E}_s\left[\sum_{t\in\mathcal{T},i\in\mathcal{N},\phi\in\Phi}\omega_i\dot{p}_{t,i,\phi}\right] \tag{1}$$

where $\mathbb{E}_s$ denotes the expectation that considers the random perturbation of the system state s; $\dot{p}_{t,i,\phi}$ denotes the actual restored load, and $\dot{p}_{t,i,\phi} = \nu_{t,i,\phi}\cdot p_{t,i,\phi}$, where $\nu_{t,i,\phi}$ is a binary variable indicating whether the load $p_{t,i,\phi}$ is energized ($\nu_{t,i,\phi} = 1$) or not ($\nu_{t,i,\phi} = 0$). Considering the limited generation capacity and the high generation cost, it is impractical to supply all end-use loads on distribution feeders using microgrids, hence critical loads have much higher restoration priority, i.e., ω_i, than non-critical loads.

2) Constraints of RoMF: We classify the constraints that the microgrid formation should obey into three categories: distribution system operational constraints, microgrid operational constraints, and the customer satisfaction constraint.

- *Distribution System Operational Constraints*

Unbalanced three-phase power flow equations should be satisfied.

$$p_{t,i,\phi} - jq_{t,i,\phi} = \left(\bar{V}_{t,i,\phi}\right)^*\sum_{j\in\Omega_i}\sum_{\phi'}Y_{ij,\phi\phi'}\bar{V}_{t,j,\phi'},$$
$$t\in\mathcal{T}, i\in\mathcal{N}, \phi,\phi'\in\{a,b,c\} \tag{2}$$

where $\bar{V}_{t,i,\phi}$ denotes the complex voltage at time step t at bus i for a given phase ϕ; Ω_i denotes the set of buses directly connected to bus i; $Y_{ij,\phi\phi'}$ denotes the nodal admittance matrix element corresponding to bus i (phase ϕ) and bus j (phase ϕ'). Notably, in this paper, constraint (2) is satisfied by establishing the power flow models of test systems in the OpenDSS.

Bus voltages should be maintained within acceptable operating limits.

$$V^{\min} \le V_{t,i,\phi} \le V^{\max}, t\in\mathcal{T}, i\in\dot{\mathcal{N}}, \phi\in\{a,b,c\} \tag{3}$$

where $V_{t,i,\phi}$ denotes the voltage magnitude at time step t at node i for a given phase ϕ; $V^{\min}$ and $V^{\max}$ denote the lower and upper limits of node voltage magnitude, respectively; $\dot{\mathcal{N}}$ denotes the energized node set, $\dot{\mathcal{N}}\subseteq\mathcal{N}$. In this paper, $V^{\min} = 0.95$ pu and $V^{\max} = 1.05$ pu.

Line currents should not exceed their limits.

$$I_e^{\min} \le I_{t,e,\phi} \le I_e^{\max}, t\in\mathcal{T}, e\in\dot{\mathcal{E}}, \phi\in\{a,b,c\} \tag{4}$$

where $I_{t,e,\phi}$ denotes the current magnitude at time step t on line e for a given phase ϕ; $I_e^{\min}$ and $I_e^{\max}$ denote the lower and upper limits of current magnitude on line e, respectively; $\dot{\mathcal{E}}$ denotes the energized line set, $\dot{\mathcal{E}}\subseteq\mathcal{E}$.

- *Microgrid Operational Constraints*

In the post-disruption stage, a microgrid can restore critical loads outside the microgrid with its surplus capacity via dynamic boundaries and formation. We take the master-slave control strategy [40], where the voltage and frequency of each islanded microgrid are controlled by only one DG, i.e., the master DG, and do not consider the microgrid cluster technique [41]. Assuming that a radial network structure should be maintained when disassembling the distribution network into several islanded microgrids and each microgrid is only energized by the master DG. Hence, each load is served by one DG through one path, and there is no path between any two DGs. For a distribution network with tie-lines, the topology constraint is formulated as:

$$\sum_{j\in\mathcal{M}} l_{i,j} \le 1, i\in\mathcal{N} \tag{5}$$

where $l_{i,j}$ is binary variable indicating whether there is a path between node i and node j ($l_{i,j} = 1$) or not ($l_{i,j} = 0$). For a radial distribution network, (5) can be simplified as:

$$\sum_{i,j\in\mathcal{M}} l_{i,j} = 0 \tag{6}$$

Actually, maintaining a radial structure helps to simplify many operation issues, e.g., synchronization and load sharing among microgrids, and the relay settings can be easily adjusted to protect the system from potential subsequent faults.

Furthermore, we regard DGs as emergency response resources and assume their fuels are adequate during the restoration period, and only consider their capacity constraints to keep the learning task concise.

$$p_{t,k}^{\mathrm{DG,min}} \le p_{t,k}^{\mathrm{DG}} \le p_{t,k}^{\mathrm{DG,max}}, t\in\mathcal{T}, k\in\mathcal{M} \tag{7}$$
$$q_{t,k}^{\mathrm{DG,min}} \le q_{t,k}^{\mathrm{DG}} \le q_{t,k}^{\mathrm{DG,max}}, t\in\mathcal{T}, k\in\mathcal{M} \tag{8}$$

where $p_{t,k}^{\mathrm{DG}}$ and $q_{t,k}^{\mathrm{DG}}$ denote the real and reactive power output of DG k at time step t, respectively; $p_{t,k}^{\mathrm{DG,min}}$ and $q_{t,k}^{\mathrm{DG,min}}$ denote the lower limits of real and reactive power output of DG k at time step t, respectively; $p_{t,k}^{\mathrm{DG,max}}$ and $q_{t,k}^{\mathrm{DG,max}}$ denote the upper limits. The effect of ramp rate limits has been considered in these output limits, as shown in Appendix A.

- *Customer Satisfaction Constraint*

Customer satisfaction is commonly found in power system reliability studies [42], [43], which is used to evaluate the quality of power services from the perspective of customers through reliability indices such as the customer interruption frequency and duration. In this paper, we consider the customer satisfaction constraint in terms of the interruption frequency, and assume that during the restoration period, the DSO will not shed a load to restore another when the two are of similar importance (i.e., the weighted load), and the DSO only need to consider whether the current strategy will cause the restored load at the last time step to be shed or not, regardless of all the other states before that. Because the importance of loads has been considered in (1), we can consider the customer satisfaction constraint by punishing the action of opening a closed switch.

B. MDP Formulation of RoMF

1) Preliminary: MDP is a classical formalization of sequential decision-making, its definition is given as follows:

Definition 1: A MDP is defined by a 5-tuple $\{\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \gamma\}$, where $\mathcal{S}$ is a state space; $\mathcal{A}$ is an action space; $\mathbb{P}(s' \mid s, a)$ is the probability of reaching state s' while performing action a in state s; $r(s, a, s')$ is the scalar reward associated to the transition from s to s' with action a; γ is a discount factor determining the agent's horizon, $\gamma \in [0, 1]$.

MDP learns from the SAEI to achieve a goal, where at time step t, the agent selects an action $a_t \in \mathcal{A}$ based on the environment state $s_t \in \mathcal{S}$, then the environment reaches s_{t+1} according to the probability $\mathbb{P}(s_{t+1} \mid s_t, a_t)$, and the agent receives a reward $r_t = r(s_t, a_t, s_{t+1})$. The agent's goal is to maximize its value in a given state s or state-action pair (s, a), which can be estimated by the expected cumulative reward. The action-value (also known as state-action-value) function $Q_\pi(s, a)$ is defined as:

$$Q_\pi(s, a) = \mathbb{E}_{\tau \sim \pi}\left[G_{\tau_t} \mid s_t = s, a_t = a\right], \forall s \in \mathcal{S}, a \in \mathcal{A} \quad (9)$$

where π denotes a mapping from states to probabilities of selecting each possible action; $Q_\pi(s, a)$ denotes the expected discounted return starting from state s, taking action a, and following policy π thereafter; τ denotes a trajectory of states, actions, and rewards; G_{τ_t} denotes the discounted return along an episode, which is a sub-trajectory starting at step t and ending at step T, $G_{\tau_t} = \sum_{k=t}^{T-1} \gamma^{k-t} r_k$. The discount factor γ indicates how many steps are considered in the return. For the RoMF problem, it refers to how many future restoration steps are considered in the current step. For example, the RoMF is reduced to a one-shot problem when $\gamma = 0$, but a multi-step (about one hundred) problem when $\gamma = 0.99$.

Then, we formulate the RoMF problem as a MDP by identifying corresponding MDP elements, including agent, environment, action, state, and reward.

2) Agent and Environment: For the RoMF problem, the agent is identified as the DSO from a centralized perspective, whose function is to develop formation strategies based on partially observable system states. The environment is identified as a distribution system simulator capable of action execution, state analysis, and reward generation. In this paper, we build an OpenDSS-based RoMF environment, and the details of its implementation is introduced in the subsequent subsection.

3) Action: The action at time step t is formulated as below:

$$a_t = \left\{o_{t,w} \mid t \in \mathcal{T}, w \in \mathcal{W}\right\} \quad (10)$$

where $o_{t,w}$ is a binary variable denoting the operation of switch w at time step t. To be specific, $o_{t,w} = 0$ for the close operation, and $o_{t,w} = 1$ for the open operation. Notably, since we adopt a simulator-based environment, and the DGs perform as voltage sources during power flow calculations, their outputs are determined as power flow calculation results rather than given in the action. The action space $\mathcal{A}$ is discrete since it is composed of limited binary variables.

4) State: The system state at time step t is formulated as below:

$$s_t = \Big\{p_{t,i,\phi}, q_{t,i,\phi}, p_{t,k}^{\mathrm{DG,min}}, p_{t,k}^{\mathrm{DG,max}}, q_{t,k}^{\mathrm{DG,min}}, q_{t,k}^{\mathrm{DG,max}}, o'_{t,w} \mid t \in \mathcal{T}, i \in \mathcal{N}, \phi \in \Phi, k \in \mathcal{M}, w \in \mathcal{W}\Big\} \quad (11)$$

where $o'_{t,w}$ is a binary variable indicating the state of switch w at time step t. To be specific, $o'_{t,w} = 1$ for the open state and $o'_{t,w} = 0$ for the closed state. We can find that $o'_{t,w} = o_{t-1,w}$ when the action takes effect. The state space $\mathcal{S}$ is continuous since it comprises continuous variables such as real and reactive power. Actually, s_t only comprises partially system parameters. For the impact of other parameters (e.g., bus voltage, line current, and load fluctuation) on the microgrid formation, we assume that the agent can learn it actively from the SAEI.

TABLE I
CATEGORIES OF CONSTRAINT VIOLATIONS

Constraints	Hard	Soft
power balance constraint	✓	
bus voltage constraint*		✓
line current constraint	✓	
topological constraint	✓	
microgrid capacity constraint	✓	
customer satisfaction constraint		✓

* A voltage offset of more than ±0.1 pu will also cause restoration failure.

5) Reward: The decision-making ability of the agent arises from the pursuit of a reward that is designed specifically to elicit that ability [44]. Hence, both the restoration object and constraints should be carefully considered when designing the reward function. The constraints discussed in the previous subsection are classified into two categories: hard and soft constraints, as shown in Table I. Furthermore, it is considered that the restoration process will fail due to hard constraint violations, but continue with soft constraint violations. The criterion to classify a constraint violation as hard or soft is whether it is impossible or unacceptable. If it is, then it is considered as hard, otherwise, it is considered as soft. Notably, both hard and soft constraint violations will cause penalties in the reward function, and further lead the agent to avoid the violations.

The reward function is formulated as:

$$r_t = \begin{cases} -c_0 & \text{for hard violations} \\ c_1\alpha_{1,t} - c_2\alpha_{2,t} - c_3\alpha_{3,t} & \text{for others} \end{cases} \quad (12)$$

$$\alpha_{1,t} = \sum_{i \in \mathcal{N}, \phi \in \Phi} \omega_i \dot{p}_{t,i,\phi} \quad (13)$$

$$\alpha_{2,t} = \sum_{i \in \mathcal{N}, \phi \in \Phi} \Big[\max\big(0, V_{t,i,\phi} - V^{\max}\big) + \max\Big(0, V^{\min} - V_{t,i,\phi}\Big)\Big] \quad (14)$$

$$\alpha_{3,t} = \sum_{s \in \mathcal{S}} \max\Big(0, o_{t,w} - o'_{t,w}\Big) \quad (15)$$

where $\alpha_{1,t}$ denotes the sum of restored loads at time step t; $\alpha_{2,t}$ denotes the bus voltage violation; $\alpha_{3,t}$ denotes the customer satisfaction violation; c_0, c_1, c_2, and c_3 are positive coefficients. In order to punish the actions that violate hard constraints, the value of c_0 is much larger than others.

C. Simulator-Based Environment

In the SAEI, the environment needs to accept the action from the agent and then feedback the system state and the reward. To achieve that, we should guarantee that all parameter related to the action, state, and reward are available, and

Algorithm 1 Environment Implementation

```
Build a distribution system in the OpenDSS and save its topology data in .json format.
Receive an action $a_t$ from the agent.
if actions violating the topology constraint are not removed in advance then
    Perform topology analysis based on graph theory.
end if
if violate the topology constraint (5) or (6) then
    Let $s_{t+1}$ be the terminal state, and $r_t = -c_0$.
else
    Execute $a_t$ and then perform power flow calculations.
    Read and analyze the power flow results.
    if violate hard constraints then
        Let $s_{t+1}$ be the terminal state, and $r_t = -c_0$.
    else
        Form $s_{t+1}$ according to (11);
        calculate $r_t$ according to (12)-(15).
    end if
end if
Feedback $s_{t+1}$ and $r_t$ to the agent.
```

all constraint violations can be checked in the environment. In general, two basic functions are required in the RoMF simulation environment: topology analysis and unbalanced three-phase power flow analysis. Topology analysis is used to check the radial topology constraint violation. Power flow analysis plays as the core to generate the state and reward, and other constraint violations can be checked based on power flow results. For example, the bus voltage violation is checked by comparing voltage results with corresponding limits.

In this paper, the environment is implemented in a simulator-based style, where the OpenDSS, a comprehensive electrical system simulation tool for electric utility distribution systems [45], is used to perform unbalanced three-phase power flow calculations. The OpenDSS provides rich interfaces that can be used to interact with other software such as Python and MATLAB. It means that the SAEI can be easily expanded by customizing or embedding other software to include more features. The pseudocode of environment implementation is shown in Algorithm 1. Firstly, the power flow models of distribution systems to be analyzed are built in the OpenDSS, where the DGs are modeled as constant voltage sources and their outputs will be given as power flow results. In addition, the graph of the distribution network is built and saved for the topology analysis. Then, after we receive the action a_t from the agent, if the actions that violate the topology constraint are not removed in advance,[3] by searching for the paths between load nodes and source nodes based on deep first search [46], the radial topology constraint (5) can be checked. Notably for a radial distribution network, the radial topology constraint (6) is used and checked by searching for the paths between source nodes. Then, the action is executed and the power flow calculation is performed in the OpenDSS. Once the topology analysis and power flow calculations are finished, we will read the results (e.g., bus voltage, line current, outputs of DGs, actual loads, switch states, etc.), based on that constraint violations can be determined, and s_{t+1} and r_{t+1} can be obtained using (11)-(15) and then fed back to the agent.

Furthermore, the implemented environment is integrated into the OpenAI Gym that provides a standard agent-environment interface for DRL tasks. By doing so, we can easily compare the proposed method with other learning-based methods.

III. Algorithm and Framework

RL is a framework for solving decision-making problems that can be represented as MDPs. In Section II, we have formulated the RoMF problem as a MDP, which has a continuous state space and a discrete action space. On this basis, we use the DQN to search for optimal policies of the RoMF without incorporating additional prior knowledge.

A. DQN and Its Training

Using value functions to organize and structure the search for good policies is a key idea of RL. The optimal action-value function is defined by:

$$Q_*(s, a) = \max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[G_{\tau_t} \mid s_t = s, a_t = a\right] \tag{16}$$

which obeys the Bellman equation. Then, the optimal strategy is to select the action a' maximizing the expected value of $r + \gamma Q_*(s', a')$, formulated as:

$$Q_*(s, a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q_*\left(s', a'\right) \mid s_t = s, a_t = a\right] \tag{17}$$

The action-value function can be estimated by using the Bellman equation as an iteration update:

$$Q_{i+1}(s, a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q_i\left(s', a'\right) \mid s_t = s, a_t = a\right] \tag{18}$$

Such value iteration algorithms converge to the optimal action value function, $Q_i \to Q_*$ as $i \to \infty$. A neural network function approximator with weight θ, referred as a Q-network and formulated as $Q(s, a; \theta) \approx Q_*(s, a)$, is used to estimate the action value function. The Q-network has observations as inputs and action-values as outputs.

The Q-network can be trained by adjusting the parameters θ_i at iteration i to reduce the mean-squared error in the Bellman equation, where the optimal target values $r + \gamma \max_{a'} Q_*(s', a')$ are substituted with approximate target values $y = r + \gamma \max_{a'} Q(s', a'; \hat{\theta}_i)$, using parameters $\hat{\theta}_i$ from some previous iteration. The loss function $L(\theta_i)$ that changes at each iteration i is formulated as:

$$L(\theta_i) = \mathbb{E}_{s,a,r,s'}\left[(y - Q(s, a; \theta))^2\right] \tag{19}$$

By differentiating the loss function with respect to the weights, we arrive at the following gradient:

$$\nabla_{\theta_i} L(\theta_i) = \mathbb{E}_{s,a,r,s'}\left[(y - Q(s, a; \theta))\nabla_{\theta_i} Q(s, a; \theta_i)\right] \tag{20}$$

Then, the mini-batch stochastic gradient descent (SGD) can be used to optimize the loss function and update the weight

[3] Since DRL algorithms are not suitable for dealing with scenarios with high failure probability, we can use the topology analysis module to identify actions that violate the topological constraint and remove them from the action space in advance to stabilize the training process of DRL agents.

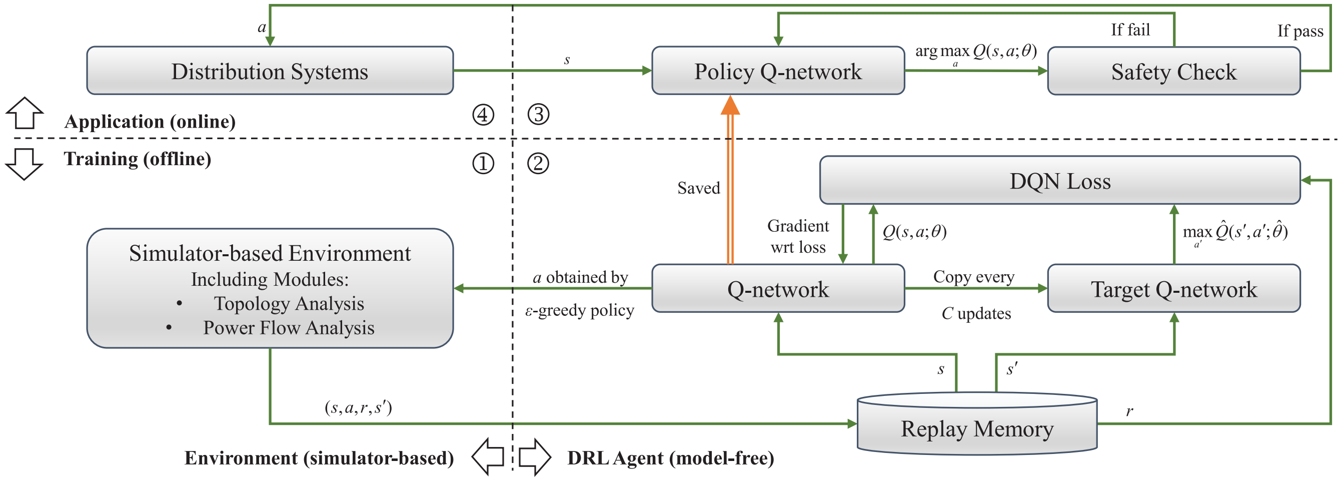


Fig. 2. Training-application framework of the DRL-based RDN-oriented microgrid formation.

θ. The algorithm of training deep Q-networks, i.e., the deep Q-learning, is presented in Appendix B.

B. Offline-Training and Online-Application

The offline-training and online-application framework of the DRL-based RoMF is shown in Fig. 2. According to training or application, environment or agent, the whole framework is divided into four parts, denoted by ① to ④ in Fig. 2. In part ①, the simulator-based environment is used to interact with the agent, i.e., feedback the experience (s, a, r, s') after accepting the agent's action. Part ② denotes the training process of the DQN, where experiences are stored as replay memory and mini-batch SGD is applied to experience samples. Every C updates the Q-network is cloned to abtain the target Q-network that is used to generate the target values. The training is terminated after a specified number of training steps, which is determined experimentally. Then, the trained Q-network is saved as the policy Q-network and used for application. In part ③, the state of the distribution system is transferred into the input layer of the policy Q-network, and the action with the highest value in the output layer is regarded as the optimal action. In addition, to address the black-box issue of deep learning, a safety check mechanism is added in the application process. The DRL-based action will be checked, and only the action that does not violate hard constraints will be applied in part ④, which refers to the microgrid formation operation in the application process. If an action cannot pass the safety check, the action with the highest value among the remaining actions will be selected and checked again.

Furthermore, by separating training and application, we can continuously improve the agent's decision-making ability with an offline style, and the agent with better performance is available for online application. Notably, by transferring massive calculations to the offline-training section, this framework has a very concise application mode, i.e., the input-output mapping. Hence, the application efficiency can be extremely advantageous, which makes the proposed method effective for online solving. Moreover, we believe that this framework is of value to other tasks in power system with machine learning applications.

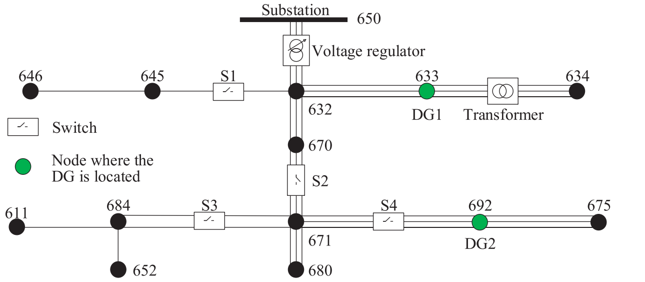


Fig. 3. Three-phase network diagram of the modified 13-bus feeder, where node 670 is an additional node connected with the distributed loads between nodes 632 and 671.

IV. Case Studies

In this section, the proposed method is validated via three modified IEEE test feeders, i.e., IEEE 13-bus feeder [47], IEEE 37-bus feeder [48], and IEEE 123-bus feeder [49], based on three-phase analysis. Python 3.6.12 and PyTorch 1.8.1 with CUDA 10.2 are used to solve the DRL-based problem. In addition, the mixed-integer linear program (MILP) method in [2] is used as a comparison, and solved with MATLAB 2021b and CPLEX 12.8.0. All experiments were performed on a personal computer with Intel Core i7-8700 CPU at 3.20 GHz, 8.00 GB RAM, and NVIDIA GeForce GT 1030.

A. Setting of Test Systems

1) 13-Bus Feeder: The three-phase diagram of the modified 13-bus feeder is shown in Fig. 3. There are 2 DGs and 4 switches that are used to conduct the microgrid formation. The load data and microgrid capacity data are shown in Tables II and III, respectively. The capacitor at node 675 is adjusted from 200 kVar per phase to 100 kVAr per phase. Other parameters are consistent with [47]. The system state of the 13-bus feeder comprises 42 parameters, including 30 load parameters (i.e., the nonzero real and reactive loads of all nodes) where

TABLE II
BASIC CRITICAL LOAD DATA OF MODIFIED IEEE 13-BUS FEEDER

Node	ω_i	Ph-1	Ph-1	Ph-2	Ph-2	Ph-3	Ph-3
		kW	kVAr	kW	kVAr	kW	kVAr
634	2	80	55	60	45	60	45
645	1	0	0	170	125	0	0
646	1	0	0	230	132	0	0
652	2	64	43	0	0	0	0
670	1	17	10	66	38	117	68
671	2	50	30	50	30	50	30
675	4	120	47	17	15	72	53
692	2	0	0	0	0	85	75
611	2	0	0	0	0	85	40

TABLE III
MICROGRID CAPACITY DATA OF MODIFIED IEEE 13-BUS FEEDER

Microgrid	Real Power (kW)	Reactive Power (kVAr)
1	[0, 930]	[-550, 550]
2	[0, 570]	[-300, 300]

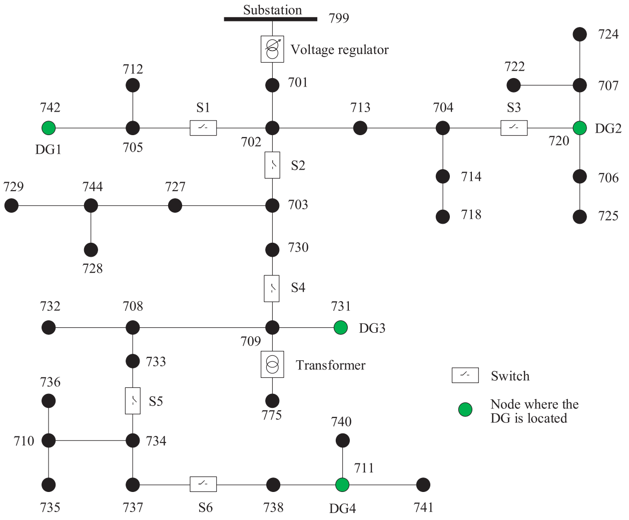


Fig. 4. Single-line network diagram of the modified 37-bus feeder.

node 671 is represented by its total real and reactive power since its three-phase load is balanced. Considering the fewer switches, the action space includes all possible switch operation combinations. Thus, the size of the action space at each step is 16.

2) 37-Bus Feeder: The single-line diagram of the modified 37-bus feeder is shown in Fig. 4. It includes 4 DGs and 6 switches to conduct the microgrid formation. Its load data and microgrid capacity data are shown in Tables IV and V, respectively. Other parameters are consistent with [48]. With this setting, the system state of the 37-bus feeder comprises 82 parameters, and the size of the action space at each step is 30.

3) 123-Bus Feeder: The single-line diagram of the modified 123-bus feeder is shown in Fig. 5. There are 8 DGs and 17 switches, including 2 tie-line switches, that are used to conduct the microgrid formation. There are six line switches in open status (at fault). The DG capacity data are shown in Table VI. The loads and other parameters are consistent with [49]. Furthermore, the system state of the 123-bus feeder comprises 225 parameters, and the size of the action space at each step is 576.

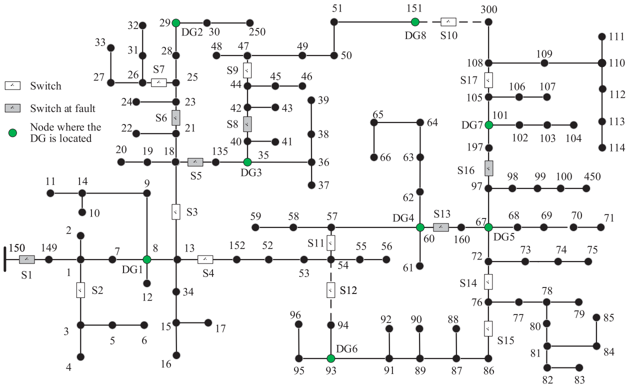


Fig. 5. Single-line network diagram of the modified 123-bus feeder. The tie-lines are denoted by the dashed lines. S10 and S12 are tie-line switches that are normally open. There are six switches are in open status (at fault).

TABLE IV
BASIC CRITICAL LOAD DATA OF MODIFIED IEEE 37-BUS FEEDER

Node*	Ph-1	Ph-1	Ph-2	Ph-2	Ph-3	Ph-3
	kW	kVAr	kW	kVAr	kW	kVAr
701	30	10	30	10	50	25
722	0	0	40	20	21	10
742	88	44	100	50	0	0

* Other node loads are consistent with the standard test system [48]. For simplification, it is considered that all critical loads have the same weight, i.e., 1.

TABLE V
MICROGRID CAPACITY DATA OF MODIFIED IEEE 37-BUS FEEDER

Microgrid	Real Power (kW)	Reactive Power (kVAr)
1~4	[0, 700]	[-400, 400]

TABLE VI
MICROGRID CAPACITY DATA OF MODIFIED IEEE 123-BUS FEEDER

Microgrid	Real Power (kW)	Reactive Power (kVAr)
1, 5	[0, 500]	[-250, 250]
2, 4	[0, 500]	[-300, 300]
3, 7	[0, 400]	[-200, 200]
6, 8	[0, 850]	[-500, 500]

Considering that the topology constraint violations can be easily checked with the topology analysis module, the action space of 37-bus and 123-bus feeders does not include the switch operations that violate topology constraint, in order to stable the training and improve sampling efficiency. The unbalanced three-phase power flow models of test systems are built based on the OpenDSS. In addition, we assume that the load at the next time step has a ±30% random fluctuation based on the current load, and the ramp rate of microgrids is 100%. For the MILP method in [2], where a single-phase power flow model is used, its load at each node is the sum of three-phase loads.

TABLE VII
LIST OF HYPERPARAMETERS AND THEIR VALUES

Hyperparameter	Value
minibatch size	64
epoch size	2,000
training steps	500,000
target network update frequency	2,000
upper limit of episode length	50
replay memory size	100,000
replay start size	2,000
initial exploration rate	0.9
final exploration rate	0.1
decay coefficient of exploration	20,000
discount factor	0.99
learning rate	0.00025

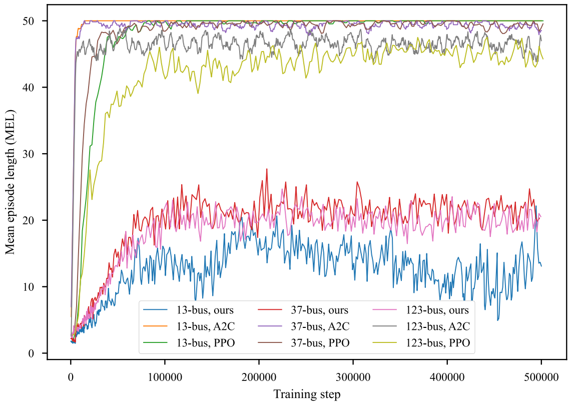


Fig. 6. Training curves of the agent's mean episode length.

B. Training Performance

In the training process, by trial and error, the structure of Q-networks is determined as a neural network consisting of four fully-connected linear layers. The neurons of all layers are followed by rectified linear units (ReLU). The number of neurons in the input layer equals to the number of state parameters. Each hidden layer includes 512 rectifier neurons. The number of neurons in the output layer equals to the size of the action space at each step. The hyperparameters required for training Q-networks are listed in Table VII, and the Adam [50] is adopted as the SGD algorithm. Furthermore, by integrating the environments into the OpenAI Gym, two reliable DRL algorithms implemented in the Stable Baselines3 (SB3) [51], i.e., advantage actor critic (A2C) and proximal policy optimization (PPO), are used as comparative learning-based algorithms.

There are two indices, i.e., mean episode length (MEL) and mean episode return (MER), that are used to describe whether the agent can improve its control policy during the agent-environment interaction. Their definitions are given in Appendix C. The training performance is illustrated by their temporal evolution, as shown in Figs. 6 and 7. We can find that the two training indices of ours are lower than the other two methods, which is expected considering the exploration mechanism of DQN. DQN allows the agent to randomly select actions with a decaying probability ϵ ($\epsilon_\infty = 0.1$) in the training process, which may result in the episode to terminate with a low return due to hard constraint violations.

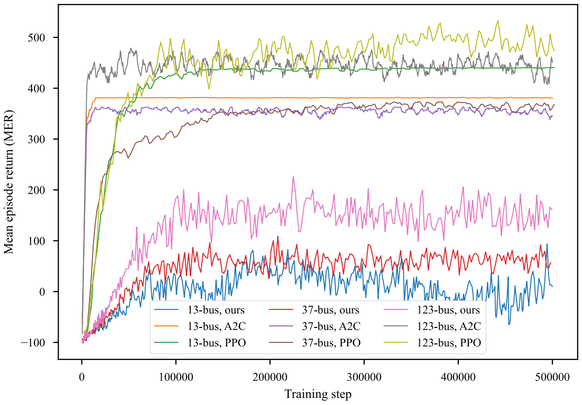


Fig. 7. Training curves of the agent's mean episode return.

Notably, this difference does not affect the final application performance because a deterministic control policy is adopted in the application process. In addition, the MEL of A2C and PPO converges to the upper limit quickly, but PPO scores higher than A2C on the MER, which indicates that PPO can learn better control strategies than A2C.

In terms of the proposed method, the evolution curve of MER can be analyzed from two factors: the episode's length and the transition's reward. Its early convergence means that the agent can quickly learn a decent control policy. After that, the agent tends to take some actions with higher rewards. But higher rewards come with higher risks, that may shorten the episode's length, also result in negative rewards. Hence, when the risk-reward tradeoff is not coordinated well, the agent may be too conservative or aggressive. The risk-reward tradeoff also applies to MEL. Furthermore, by monitoring the output of the Q-network, we found that the temporal evolution of action-value can be divided into two categories. One converges to their expected discount return, which corresponds to legal actions. The other converges to zero,[4] which corresponds to illegal actions (e.g., the actions that violate the topology constraint for the 13-bus feeder), indicating that hard constraints can be effectively learned.

In addition, the training performance is sensitive to the setting of the action space. The action space is relatively small for the 13-bus feeder. But its illegal actions account for more since it use a full action space. For the other two feeders, although their action space is larger, but their illegal actions account for less, which results in a better training performance. In general, the action space at each step should not be too large, and using specific rules to reduce the proportion of illegal actions, e.g., eliminating the actions that violate the topological constraint, can be effective to stable the training.

The result of training efficiency is shown in Table VIII. It takes several hours to train an agent, and the training efficiency of different DRL algorithms is similar.

[4] According to the definition of reward, the value of illegal actions (i.e., actions that inevitably violate hard constraints) should be negative. However, the neurons of the output layer are followed by ReLU, whose formulation is $f(x) = \max(0, x)$, so the value of illegal actions converges to zero instead of negative values.

TABLE VIII
COMPUTATIONAL EFFICIENCY OF TRAINING*

Method	13-bus	37-bus	123-bus
Ours (h)	2.08	2.84	5.49
A2C (h)	2.92	3.49	6.30
PPO (h)	2.72	3.42	6.18

* The computational efficiency is GPU-based, which provides about 4% acceleration over the CPU-based training for this experiment.

TABLE IX
APPLICATION INDICES

Method	Index	13-bus	37-bus	123-bus
Ours	ASR	99.40%	99.98%	99.66%
	ASR-*sub**	100%	100%	100%
	AOG	2.09%	0.01%	1.63%
A2C	ASR	100%	100%	99.91%
	AOG	22.26%	3.08%	31.60%
PPO	ASR	100%	100%	99.79%
	AOG	6.47%	0.57%	5.21%

* *sub* denotes the using of sub-optimal solutions.

C. Application Performance

The application performance of trained agents is tested in this subsection. By virtue of the input-output mapping of the neural network, i.e., feeding the system state into the neural network and selecting the optimal action based on the outputs, we can easily get a RoMF solution. Two application indices are used to measure the application performance, i.e., average success rate (ASR) and average optimality gap (AOG). Their definitions are given in Appendix C. ASR denotes the probability that the agent taking a legal action, mainly used to address the black box characteristics of neural networks. AOG measures the proximity of the DRL-based solution and the benchmark solution in terms of the restoration target (i.e., (1)). The closer it is, the better the performance of the agent on decision optimality.

The benchmark solution for an episode is obtained by solving a sequence of one-shot static CLR problems. Each one-shot static CLR process, corresponding to a transition in the episode, is solved by executing all possible actions in turns in the OpenDSS-based environment, and the action with maximum restored load is regarded as the optimal solution. By doing so, the size of action space of an episode of length T is simplified from n^T to $n \cdot T$ where n is the number of actions at each step. Since the benchmark method neglects the constraints between transitions, its solution might be over-optimistic. Hence, we take it as a reference and believe that the closer the learning-based solution is to the reference (the proximity between the two is described by the index AOG), the better the performance of the learning-based method.

The application results are shown in Table IX. For the proposed method, the ASR reaches 99.40%, 99.98% and 99.66% for 13-bus, 37-bus and 123-bus feeders, respectively. Moreover, the safety check mechanism is considered in the ASR-sub. To be specific, the sub-optimal solution (i.e., the action with the second highest output value) will be selected if the optimal solution violates hard constraints, and at most one reselection can be performed in one transition. With this setting, the ASR-sub is basically 100% for all test feeders. In addition, A2C and PPO have higher ASR but worse AOG. It actually means that the agents of A2C and PPO are too conservative.

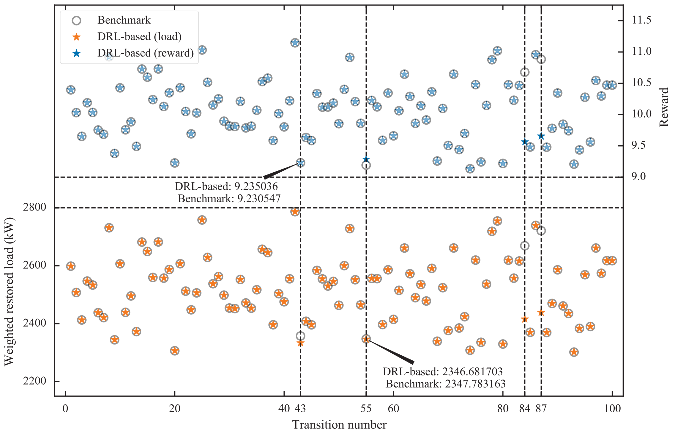
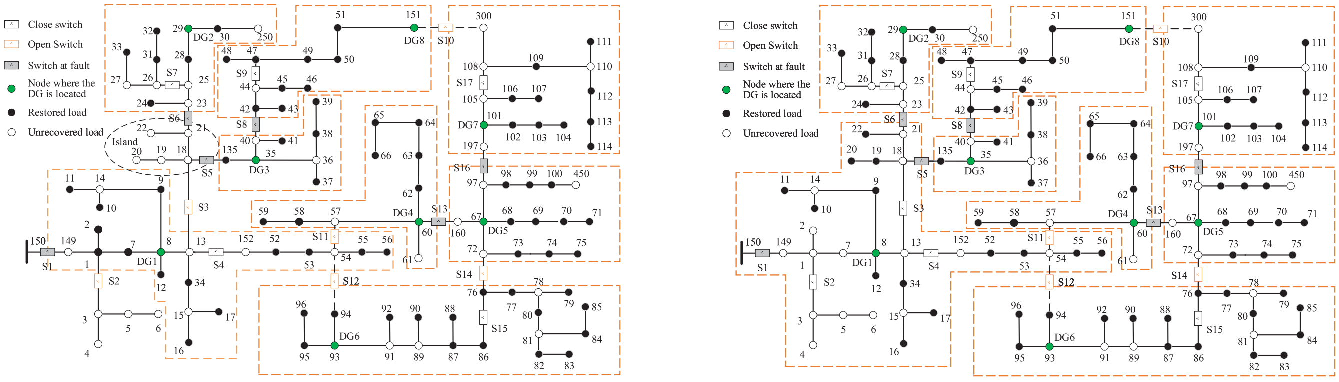


Fig. 8. Comparison of DRL-based solutions and benchmark solutions, including the temporal evolution of the immediate reward and the weighted restored load.

A pair of DQN-based solutions and benchmark solutions of the 13-bus feeder are used to illustrate the reason for the optimality gap, as shown in Fig. 8. We can find that the optimality gap mainly comes from two aspects: 1) The agent will make some conservative decisions, e.g., the 84th and 87th transitions in Fig. 8. 2) After considering the customer satisfaction constraint, the agent will make some decisions with higher immediate rewards but smaller restored load, e.g., the 43th and 55th transitions in Fig. 8. The former reflects agent's risk-averse characteristics, while the latter exactly embodies its effectiveness of dealing complex constraints.

Furthermore, the MILP method in [2] is used as a comparison to illustrate the differences in decision-making features and computational efficiency between the proposed method and the MP-based method. Fig. 9 shows the one-shot microgrid formation results for the 123-bus feeder obtained by the proposed method and the MILP method. Eight microgrids are formed separately by operating switches, and each one is energized by one DG. The voltage of nodes are within the range specified. The loads picked up are labeled by solid circles in Fig. 9. The MILP method in [2] has a similar restoration object and constraints with the proposed method, but it assumes that all islands caused by the failure before recovery have been identified and isolated, and all loads are controllable and no new islands will be formed during the recovery process. Such assumptions are common in MP-based works to make mathematical models normative. However, these feature are actually allowed in power flow calculations. By learning from the interaction with the simulator-based environment, the proposed method allows the existence of uninterruptible loads and the new islands caused by the microgrid formation, as shown in Fig. 9. The proposed method provides a promising way to address more complex objects by integrating more features into the simulator-based environment. As for the MP-based



(a)

(b)

Fig. 9. Comparison of microgrid formation results for the 123-bus feeder. (a) denotes the microgrid formation results obtained by the proposed method. (b) denotes the microgrid formation results obtained by the MILP method in [2].

TABLE X
COMPUTATIONAL EFFICIENCY OF APPLICATION

	13-bus	37-bus	123-bus
Ours (s)	0.97×10^{-4}	0.99×10^{-4}	0.27×10^{-3}
A2C (s)	0.75×10^{-3}	0.76×10^{-3}	1.33×10^{-3}
PPO (s)	0.77×10^{-3}	0.78×10^{-3}	1.33×10^{-3}
Benchmark (s)	0.15	0.43	19.20
MILP [2] (s)	0.13	0.22	1.87
Benchmark/DQN	1,546	4,343	71,111
MILP/DQN	1,340	2,222	6,926

method, by establishing an accurate mathematical model, a richer action space can be considered.

Moreover, the application efficiency for different methods is shown in Table X. The efficiency denotes the average time taken per decision-making. We can find that the proposed method provides near-zero time-consuming application efficiency, which is thousands of times faster than the benchmark method and the MILP method. The difference between the proposed method and other learning-based methods is mainly resulted by the different ways of saving and calling neural networks. Moreover, with similar neural network architectures, the efficiency of learning-based methods is less affected by the scale of the test system. Notably, the benchmark and MILP methods are both one-shot style, and hard or even impossible to solve dynamic problems that include numerous scenarios.

V. CONCLUSION

In this work, we propose a novel resilience-oriented microgrid formation scheme based on DRL techniques. By interacting with a simulator-based environment, the proposed method supports a model-free decision-making manner. Case studies have demonstrated that, with will-tuned hyperparameters, the proposed DRL-based method can effectively learn control policies without incorporating additional prior knowledge. In the application process, the decision success rate can be effectively guaranteed by incorporating a safety check mechanism. In general, the proposed method provides near-optimal solution with near-zero time-consuming application efficiency.

We also realize that this work still has many limitations. Currently, we are working on using multi-agent reinforcement learning to deal with large-scale distribution networks and complex restoration actions. Furthermore, we plan to develop the resilience-oriented microgrid formation environment on open-source platforms, and new features such as black-start DGs, transient features of DGs, fuel adequacy and renewable energy resources may be included. We think that interesting future research directions open up for this work.

APPENDIX A
FORMULATIONS OF DG OUTPUT LIMITS

$$p_{t,k}^{DG,\min} = \max\left(p_k^{DG,\min}, p_{t-1,k}^{DG} - \delta_k^p p_k^{DG,\max}\right) \tag{21}$$

$$p_{t,k}^{DG,\max} = \min\left(p_k^{DG,\max}, p_{t-1,k}^{DG} + \delta_k^p p_k^{DG,\max}\right) \tag{22}$$

$$q_{t,k}^{DG,\min} = \max\left(q_k^{DG,\min}, q_{t-1,k}^{DG} - \delta_k^q q_k^{DG,\max}\right) \tag{23}$$

$$q_{t,k}^{DG,\max} = \min\left(q_k^{DG,\max}, q_{t-1,k}^{DG} + \delta_k^q q_k^{DG,\max}\right) \tag{24}$$

where $p_k^{DG,\min}$, $p_k^{DG,\max}$, $q_k^{DG,\min}$, and $q_k^{DG,\max}$ denote the capacity limits of DG k; δ_k^p and δ_k^q respectively denote the ramp rate of real and reactive power of DG k.

APPENDIX B
TRAINING ALGORITHM

The training algorithm is shown in Algorithm 2, where the agent selects and executes actions according to the following ϵ-greedy policy [38]:

$$\epsilon = \epsilon_\infty + (\epsilon_0 - \epsilon_\infty)e^{-t_\epsilon/c_\epsilon} \tag{25}$$

where ϵ_0 and ϵ_∞ denote the initial and final exploration rate, respectively; t_ϵ denotes the transition step; c_ϵ denotes the decay coefficient of exploration.

Two tricks are used to make this algorithm suitable for training large neural networks without diverging, i.e., experience replay and target network [38], [39]. In addition, the failure probability of the microgrid formation is not so low that the memory pool may contain a proportion of negative experiences

Algorithm 2 Deep Q-Learning

```
Initialize replay memory $\mathcal{D}$ to the replay start size.
Initialize action value function $Q$ with random weights $\theta$ and target action value function $\hat{Q}$ with weights $\hat{\theta} = \theta$.
for episode = 1, $H$[5] do
    $s \leftarrow s_1$ Initialize the environment.
    for $t$ = 1, $T$ do
        With probability $\epsilon$ select a random action $a_t$.
        Otherwise select $a_t = \text{argmax}_a Q(s_t, a; \theta)$.
        Execute action $a_t$ in simulator and observe reward $r_t$ and state $s_{t+1}$.
        Store transition $(s_t, a_t, r_t, s_{t+1})$ in $\mathcal{D}$.
        Sample a random minibatch of transitions $\eta_j$.
        if all $s_{j+1}$ in the minibatch are terminals then
            Resample a minibatch of transitions from $\mathcal{D}$.
        end if
        if episode terminates at step $j + 1$ then
            $y_j = r_j$
        else
            $y_j = r_j + \gamma \max_{a'} \hat{Q}(s_{j+1}, a'; \hat{\theta})$
        end if
        Perform gradient descent on $(y_j - Q(s_j, a_j; \theta))^2$ with respect to the network parameters $\theta$.
        Every $C$ steps reset $\hat{Q} = Q$.
    end for
end for
```

(i.e., experiences with terminal states as next-states). Hence, to avoid that the training is terminated prematurely due to the sampling of a mini-batch of all negative experiences, a resampling technique is developed, as shown in Algorithm 2.

Appendix C
Definitions of Evaluation Indices

A. Training Indices

$$\text{MEL} = \frac{1}{n_{\text{eps}}} \sum_{1}^{n_{\text{eps}}} T_i \tag{26}$$

$$\text{MER} = \frac{1}{n_{\text{eps}}} \sum_{i=1}^{n_{\text{eps}}} \sum_{j=1}^{T_i} r_{i,j} \tag{27}$$

where T_i denote the length of episode i; n_{eps} denotes the number of episodes.

B. Application Indices

For ASR, we invoke the Q-network to generate 10 episodes with maximum length not exceeding 100. The length limit is set to avoid generating excessively long episodes, which is time-consuming. Then, ASR is defined as the proportion of legal actions in all episodes, and formulated as:

$$\text{ASR} = 1 - \frac{\sum_i F_i}{\sum_i T_i} \tag{28}$$

where F_i is an indicator denoting whether the i-th episode is ended with an illegal action ($F_i = 1$) or not ($F_i = 0$).

For AOG, we invoke the Q-network to generate 10 episodes without failed transition, and their length is set to 100. Then, AOG is defined as the average optimality gap between the DRL-based method and the benchmark method, and formulated as:

$$\text{AOG} = \frac{1}{10} \sum_{i=1}^{10} \left(1 - \frac{\sum_{t \in \mathcal{T}, j \in \mathcal{N}, \phi \in \Phi} w_j \dot{p}_{t,j,\phi,i}}{\sum_{t \in \mathcal{T}, j \in \mathcal{N}, \phi \in \Phi} w_j \dot{p}^{\text{BM}}_{t,j,\phi,i}} \right) \tag{29}$$

where $\dot{p}^{\text{BM}}_{i,t,j,\phi}$ denotes the actual restored load in the benchmark solution.

References

[1] S. Shetty, G. Kamdem, B. Krishnappa, and D. Nikol, "Cyber resilience metrics for bulk power system," *Risk Anal. J.*, to be published.

[2] C. Chen, J. Wang, F. Qiu, and D. Zhao, "Resilient distribution system by microgrids formation after natural disasters," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 958–966, Mar. 2016.

[3] M. Panteli, C. Pickering, S. Wilkinson, R. Dawson, and P. Mancarella, "Power system resilience to extreme weather: Fragility modeling, probabilistic impact assessment, and adaptation measures," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3747–3757, Sep. 2017.

[4] R. Rocchetta, E. Zio, and E. Patelli, "A power-flow emulator approach for resilience assessment of repairable power grids subject to weather-induced failures and data deficiency," *Appl. Energy*, vol. 210, pp. 339–350, Jan. 2018.

[5] M. Panteli and P. Mancarella, "Influence of extreme weather and climate change on the resilience of power systems: Impacts and possible mitigation strategies," *Electr. Power Syst. Res.*, vol. 127, pp. 259–270, Oct. 2015.

[6] J. Wang, W. Zuo, L. Rhode-Barbarigos, X. Lu, J. Wang, and Y. Lin, "Literature review on modeling and simulation of energy infrastructures from a resilience perspective," *Rel. Eng. Syst. Saf.*, vol. 183, pp. 360–373, Mar. 2019.

[7] D. K. Mishra, M. J. Ghadi, A. Azizivahed, L. Li, and J. Zhang, "A review on resilience studies in active distribution systems," *Renew. Sustain. Energy Rev.*, vol. 135, Jan. 2021, Art. no. 110201.

[8] M. Panteli, D. N. Trakas, P. Mancarella, and N. D. Hatziargyriou, "Power systems resilience assessment: Hardening and smart operational enhancement strategies," *Proc. IEEE*, vol. 105, no. 7, pp. 1202–1213, Jul. 2017.

[9] D. G. Photovoltaics and E. Storage, *IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces*, IEEE Standard 1547-2018, 2018.

[10] Z. Bie, Y. Lin, G. Li, and F. Li, "Battling the extreme: A study on the power system resilience," *Proc. IEEE*, vol. 105, no. 7, pp. 1253–1266, Jul. 2017.

[11] Y. Wang, A. O. Rousis, and G. Strbac, "On microgrids and resilience: A comprehensive review on modeling and operational strategies," *Renew. Sustain. Energy Rev.*, vol. 134, Dec. 2020, Art. no. 110313.

[12] Z. Wang and J. Wang, "Self-healing resilient distribution systems based on sectionalization into microgrids," *IEEE Trans. Power Syst.*, vol. 30, no. 6, pp. 3139–3149, Nov. 2015.

[13] S. Lei, C. Chen, Y. Song, and Y. Hou, "Radiality constraints for resilient reconfiguration of distribution systems: Formulation and application to microgrid formation," *IEEE Trans. Smart Grid*, vol. 11, no. 5, pp. 3944–3956, Sep. 2020.

[14] J. Zhu, Y. Yuan, and W. Wang, "An exact microgrid formation model for load restoration in resilient distribution system," *Int. J. Elect. Power Energy Syst.*, vol. 116, Mar. 2020, Art. no. 105568.

[15] A. Hussain, V.-H. Bui, and H.-M. Kim, "Microgrids as a resilience resource and strategies used by microgrids for enhancing resilience," *Appl. Energy*, vol. 240, pp. 56–72, Apr. 2019.

[16] J. Kim and Y. Dvorkin, "Enhancing distribution system resilience with mobile energy storage and microgrids," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 4996–5006, Sep. 2019.

[17] K. S. A. Sedzro, A. J. Lamadrid, and L. F. Zuluaga, "Allocation of resources using a microgrid formation approach for resilient electric grids," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 2633–2643, May 2018.

[18] J. Zhu *et al.*, "Dynamic island partition for distribution system with renewable energy to decrease customer interruption cost," *J. Elect. Eng. Technol.*, vol. 12, no. 6, pp. 2146–2156, 2017.

[19] M. A. Gilani, A. Kazemi, and M. Ghasemi, "Distribution system resilience enhancement by microgrid formation considering distributed energy resources," *Energy*, vol. 191, Jan. 2020, Art. no. 116442.

[20] Y. Bian, C. Chen, Y. Huang, Z. Bie, and J. P. S. Catalao, "Service restoration for resilient distribution systems coordinated with damage assessment," *IEEE Trans. Power Syst.*, early access, Dec. 21, 2021, doi: 10.1109/TPWRS.2021.3137257.

[21] H. Gao, Y. Chen, Y. Xu, and C.-C. Liu, "Resilience-oriented critical load restoration using microgrids in distribution systems," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2837–2848, Nov. 2016.

[22] Y. Xu *et al.*, "DGs for service restoration to critical loads in a secondary network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 435–447, Jan. 2019.

[23] Q. Zhang, Z. Ma, Y. Zhu, and Z. Wang, "A two-level simulation-assisted sequential distribution system restoration model with frequency dynamics constraints," *IEEE Trans. Smart Grid*, vol. 12, no. 5, pp. 3835–3846, Sep. 2021.

[24] Y. Xu, C.-C. Liu, K. P. Schneider, F. K. Tuffner, and D. T. Ton, "Microgrids for service restoration to critical load in a resilient distribution system," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 426–437, Jan. 2018.

[25] J. Duan *et al.*, "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 814–817, Jan. 2020.

[26] Q. Huang, R. Huang, W. Hao, J. Tan, R. Fan, and Z. Huang, "Adaptive power system emergency control using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1171–1182, Mar. 2020.

[27] B. Wang, Y. Li, W. Ming, and S. Wang, "Deep reinforcement learning method for demand response management of interruptible load," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3146–3155, Jul. 2020.

[28] R. Lu and S. H. Hong, "Incentive-based demand response for smart grid with reinforcement learning and deep neural network," *Appl. Energy*, vol. 236, pp. 937–949, Feb. 2019.

[29] J. R. Vázquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: A review of algorithms and modeling techniques," *Appl. Energy*, vol. 235, pp. 1072–1089, Feb. 2019.

[30] T. Qian, C. Shao, X. Wang, and M. Shahidehpour, "Deep reinforcement learning for EV charging navigation by coordinating smart grid and intelligent transportation system," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1714–1723, Mar. 2020.

[31] X. Qu, Y. Yu, M. Zhou, C.-T. Lin, and X. Wang, "Jointly dampening traffic oscillations and improving energy consumption with electric, connected and automated vehicles: A reinforcement learning based approach," *Appl. Energy*, vol. 257, Jan. 2020, Art no. 114030.

[32] Y. Gao, W. Wang, J. Shi, and N. Yu, "Batch-constrained reinforcement learning for dynamic distribution network reconfiguration," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 5357–5369, Nov. 2020.

[33] Q. Zhang, K. Dehghanpour, Z. Wang, and Q. Huang, "A learning-based power management method for networked microgrids under incomplete information," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1193–1204, Mar. 2020.

[34] Q. Zhang, K. Dehghanpour, Z. Wang, F. Qiu, and D. Zhao, "Multi-agent safe policy learning for power management of networked microgrids," *IEEE Trans. Smart Grid*, vol. 12, no. 2, pp. 1048–1062, Mar. 2021.

[35] M. M. Hosseini and M. Parvania, "Resilient operation of distribution grids using deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 2100–2109, Mar. 2022.

[36] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[37] G. Brockman *et al.*, "OpenAI gym," 2016, *arXiv:1606.01540*.

[38] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[39] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[40] T. Caldognetto and P. Tenti, "Microgrids operation based on master–slave cooperative control," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 2, no. 4, pp. 1081–1088, Dec. 2014.

[41] S. Parhizi, H. Lotfi, A. Khodaei, and S. Bahramirad, "State of the art in research on microgrids: A review," *IEEE Access*, vol. 3, pp. 890–925, 2015.

[42] G. Li, Z. Bie, H. Xie, and Y. Lin, "Customer satisfaction based reliability evaluation of active distribution networks," *Appl. Energy*, vol. 162, pp. 1571–1578, Jan. 2016.

[43] M. J. Sullivan, B. N. Suddeth, T. Vardell, and A. Vojdani, "Interruption costs, customer satisfaction and expectations for service reliability," *IEEE Trans. Power Syst.*, vol. 11, no. 2, pp. 989–995, May 1996.

[44] D. Silver, S. Singh, D. Precup, and R. S. Sutton, "Reward is enough," *Artif. Intell.*, vol. 299, Oct. 2021, Art. no. 103535.

[45] "Open Distribution System Simulator (OpenDSS)." Electric Power Research Institute (EPRI). Mar. 2021. [Online]. Available: https://www.epri.com/pages/sa/opendss

[46] D. B. West *et al.*, *Introduction to Graph Theory*, vol. 2. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

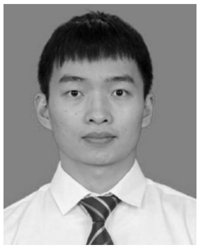
[47] "IEEE 13-Bus Feeder." IEEE PES AMPS DSAS Test Feeder Working Group. Feb. 2014. [Online]. Available: http://site.ieee.org/pes-testfeeders/files/2017/08/feeder13.zip

[48] "IEEE 37-Bus Feeder." IEEE PES AMPS DSAS Test Feeder Working Group. Feb. 2014. [Online]. Available: http://site.ieee.org/pes-testfeeders/files/2017/08/feeder37.zip

[49] "IEEE 123-Bus Feeder." IEEE PES AMPS DSAS Test Feeder Working Group. Feb. 2014. [Online]. Available: https://cmte.ieee.org/pes-testfeeders/wp-content/uploads/sites/167/2017/08/feeder123.zip

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[51] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *J. Mach. Learn. Res.*, vol. 22, no. 268, pp. 1–8, 2021.



Yuxiong Huang (Student Member, IEEE) received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2017, where he is currently pursuing the Ph.D degree. His major research interests include reliability evaluation and machine learning technologies in power systems.

Gengfeng Li (Member, IEEE) received the Ph.D. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2014, where he is currently an Associate Professor with the School of Electrical Engineering. His research interests include power system reliability evaluation, grid resilience, and integration of renewable energy.

Chen Chen (Senior Member, IEEE) received the B.S. and M.S. degrees from Xi'an Jiaotong University (XJTU), Xi'an, China, in 2006 and 2009, respectively, and the Ph.D. degree in electrical engineering from Lehigh University, Bethlehem, PA, USA, in 2013. He is currently a Professor with the School of Electrical Engineering, XJTU. Prior to joining XJTU, he has over six-year service with Argonne National Laboratory, Lemont, IL, USA, with the last appointment as an Energy Systems Scientist with Energy Systems Division. His research interest includes power system resilience, distribution systems and microgrids, demand side management, and communications and signal processing for smart grid. He was a recipient of the IEEE PES Chicago Chapter Outstanding Engineer Award in 2017. He is an Editor of IEEE TRANSACTIONS ON SMART GRID and IEEE POWER ENGINEERING LETTERS.

Yiheng Bian (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from North China Electric Power University, Baoding, China, in 2017. He is currently pursuing the Ph.D. degree with Xi'an Jiaotong University, Xi'an, China. His major research interests include planning and operation of resilient power systems.

Tao Qian (Student Member, IEEE) received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2017, where he is currently pursuing the Ph.D. degree. His current research interests include the coordinated power and traffic systems and DRL methods.

Zhaohong Bie (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Shandong University, Jinan, China, in 1992 and 1994, respectively, and the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 1998, where she is currently a Professor. Her research interests include power system reliability evaluation, integration of renewable energy, grid resilience, energy Internet, and microgrids.