Self-attention Generative Adversarial Network Enhanced Learning Method for Resilient Defense of Networked Microgrids Against Sequential Events

Jin Zhao, *Member*, *IEEE*, Fangxing Li, *Fellow*, *IEEE*, Haoyuan Sun, *Graduate Student Member*, *IEEE*, Qiwei Zhang, *Graduate Student Member*, *IEEE*, Hang Shuai, *Member*, *IEEE*

***Abstract*—The resilient responses of networked microgrids (MGs) can greatly improve the survival of critical loads during extreme events. In order to efficiently handle the scarce data issue as well as improve the adaptability of deep reinforcement learning (DRL) methods for complex sequential extreme events (SEEs) such as hurricanes and tornadoes, a new learning-based method is proposed for the survival of critical loads in MGs during SEEs. A generative adversarial network (GAN) is applied to generate a sufficient extreme event-related database in a model-free way. Specifically, a self-attention GAN (SA-GAN) is developed to capture sequential features of the SEE process. Then, the SA-GAN is integrated into a DRL framework, and the corresponding Markov decision process (MDP) and the environment are designed to realize adaptive networked MG reconfiguration for the survival of critical loads. Faced with uncertain distributed generator (DG) output and sequential line damage, the SA-GAN-DRL method provides an adaptive model-free solution to continuously supply critical loads during SEEs. The effectiveness of the proposed method is validated using a 7-bus test system and the IEEE 123-bus system, and the results demonstrate both a strong learning ability with limited practice data, and robustness and adaptability for highly changeable SEE processes.**

***Index Terms*—Self-attention, deep reinforcement learning (DRL), microgrids (MGs), distributed generator (DG), generative adversarial network (GAN).**

I. Introduction

THE extensive damage and subsequent outages within power systems caused by high-impact and low-probability extreme events indicate the necessity of enhancing power system resilience [1-2]. Networked microgrids (MGs) or multiple MGs, core components in a distributed system, are essential for enhancing the operational flexibility and efficiency of the distributed system (DS), and they offer promising solutions for power grids to withstand unplanned catastrophic events [3]. The networked MGs are self-supported MGs with the ability to provide supports to each other via reconfigurations. With their distributed but coordinated feature, they reduce the impact of cascading events and enhance the survival of smart grid critical loads. A DS with distributed energy resources (DERs) or distributed generators (DGs) can transform into a networked MG in preparation for extreme events [4]. Networked MGs can benefit from the survival of critical loads based on strategic management of local DGs as well as mutual assistance among MGs [5], [6]. Finally, networked MGs help achieve bottom-up restoration by supplying available local power sources for system restoration [7]. This paper focuses on the flexible reconfiguration of networked MGs as a resilient defense strategy during sequential extreme events (SEEs).

SEEs, e.g., hurricanes and tornadoes, pass through an area sequentially and regionally [8], and consequently lead to ongoing damage to power grid infrastructures such as generators and lines. Due to the changing environment, resilient defenses for SEEs are highly related to system conditions. Faced with subsequent damages, ref. [5] enhanced the survival of critical loads by minimizing MG scales and allowing both radial and looped MG networks. Ref. [9] provided a stochastic programming model to re-configure networked MGs with the goal of maximizing continuous load supply according to occurred fault conditions. Refs. [8] and [10] took the hurricane track as a Markov decision process (MDP), integrated it into energy dispatching models, and then applied stochastic and robust optimization methods to provide solutions, respectively. A multi-stage and multi-zone-based uncertainty set was designed for a SEE process in [11], and two-stage robust optimization was applied to minimize load shedding under extreme events.

Mathematical model-based methods use stochastic programming or robust optimization to handle uncertain conditions, while some advanced machine learning methods [12]-[20] can give solutions which are naturally adaptive for uncertain events. Convolutional neural networks (CNNs) [12] have provided satisfying model-free solutions to handle uncertain power supplies in contingency screening [13] and uncertain load restoration [14]. Regarding energy dispatch [15] and control [16] problems with dynamic Markov features, deep reinforcement learning (DRL) methods have been widely used for decision-making. Equipped with imitation learning, DRL method handled real-time service restoration issues in resilient distribution systems [17]. Ref. [18] proposed a batch constrained DRL to realize a data-driven dynamic distribution network reconfiguration. A DRL method with distributed training was applied in [19] to solve large-scale networked MG power management problems. By continuously interacting with the environment and obtaining feedback [21], the DRL method has shown the adaptability which is necessary for dealing with SEEs. However, the data under extreme events suffers scarcity issues [22]-[24]. system data from SEEs is much harder to obtain than system data from normal

This work was supported in part by NSF grant EECS-1809458. (*Corresponding author*: F. Li).

All authors are with the Department of EECS, The University of Tennessee, Knoxville. TN 37996, USA.

cases. The scarcity of SEE system data could negatively affect DRL training processes, and efficient training is critical to ensure reliable DRL applications [17]-[19]. Therefore, in order to ensure reliable implementation with limited historical/practical data, the DRL method needs further improvement.

A direct solution is to reasonably extend the original dataset. Under a SEE process, the uncertainties in renewable resources and dynamic line outage damage conditions are key to the proper resilient reactions of smart grids [8]-[10]. The SEE-related dataset contains high-dimensional information and reflects the time-varying nature of weather as well as complex energy conversion processes which make it difficult for precise analysis via model-based methods. However, generative adversarial networks (GANs) [25], as an un-supervised model-free method, can automatically extract data features without labeling. With an outstanding capability for learning data properties, GANs have been applied to generate renewable scenarios [26], distinguish power grid outage data [22] and improve event classification [27]. For the SEE-related data, GANs can extend the dataset by generating new and credible extra data sets that capture the intrinsic features of the original data. The GAN enhanced datasets can further benefit the performance of DRL.

For the purpose of enhancing resilience during a SEE process, this paper proposes a self-attention GAN enhanced DRL (SA-GAN-DRL) method for survival of critical loads by flexibly reconfiguring networked MGs. First, the Wasserstein GAN (WGAN) is applied to organize an adversarial training process for data feature extraction. Then, the self-attention GAN (SA-GAN) is further developed using *Attention Mechanism* to properly generate data with strong sequential features, e.g., line damage data. Finally, the SA-GAN is equipped into a reinforcement learning architecture called double deep Q-network (DDQN), and an environment is designed to train the SA-GAN-DDQN.

The contributions of this paper can be summarized as follows: 1) A new learning-based resilient defense scheme is proposed for networked MGs against SEEs. Accordingly, networked MGs can provide adaptive reconfiguration strategies to ensure survival of critical loads during dynamic extreme weather events. 2) The datasets of uncertain DG (wind-DG and solar DG) outputs and line damage conditions, which are of particular interests in extreme event related resilience studies, are extended to reduce data scarcity. Without any additional data analysis efforts, the originally limited historical data is reasonably and credibly extended using un-supervised model-free GAN based methods. The original GAN is further improved to formulate a customized SA-GAN which has a stronger ability to learn from sequential data in an MDP. 3) With a combination of the DRL method with GAN based data extension approach, the proposed SA-GAN-DRL method improves the application performance of original DRL algorithms. With more stable feasible actions and higher reward values, the SA-GAN-DRL has improved adaptability when applied to dynamic SEE conditions such that the resilience of networked MGs can be further enhanced during SEEs.

The rest of this paper is organized as follows: Section II introduces the adversarial nets framework of the GAN, the architecture of the developed SA-GAN, and the generative learning process of GAN method. The SA-GAN enhanced DDQN method and the designed training environment are presented in Section III for enhancing networked MG resilience. Section IV provides case study results on a 7-bus DS and the IEEE 123-bus DS with multiple MGs, followed by the conclusions in Section V.

II. GAN based Dataset Processing for Resilient Networked MGs under SEEs

GAN-based data extension is introduced in this section. First, the adversarial learning principle and structure of a classical GAN are presented. Then, the self-attention module is built and incorporated into the GAN to form the SA-GAN. Finally, the training data organization and learning algorithm are introduced.

A. Review of Wasserstein GAN

A GAN is an adversarial nets framework for estimating generative models via an adversarial process. As shown in Fig. 1, a GAN contains two deep neural networks: a generative model *Generator* that captures the data distribution, and a discriminative model *Discriminator* that estimates the probability that a sample came from the training data rather than the *Generator* [19]. For dataset extension, the goal of a GAN is to learn the features of the original dataset by figuring out a mapping relationship from a known distribution $\mathbb{P}_Z$ (such as a Gaussian distribution) to a targeted sample dataset (experience/history) that follows a distribution $\mathbb{P}_X$. The function of the GAN relies on the adversarial process, which is formulated as a game-theoretic two-player nested min-max optimization of the *Generator* and *Discriminator*. As long as the GAN is well-trained, the *Generator* is able to capture historical data features and extend the dataset for DRL training. Theoretically, a GAN can generate samples without any size limit.

The WGAN is an efficient GAN architecture that improves training stability. It provides a Wasserstein metric-based loss function to describe the quality of the generated samples [26], which helps resolve the model collapse issue. Therefore, the GAN in this paper follows the WGAN structure which is introduced below.

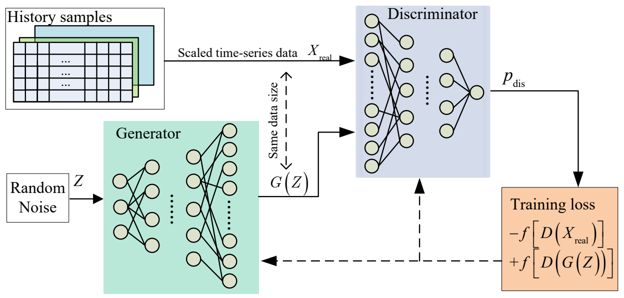

History samples
Scaled time-series data X_{real}
Discriminator
p_{dis}
Same data size
Generator
Random Noise
Z
$G(Z)$
Training loss
$-f\left[D\left(X_{\text{real}}\right)\right]$
$+f\left[D\left(G(Z)\right)\right]$

Fig. 1. Structure of GAN

1) Generator

The *Generator* is trained to transform a random noise signal Z with distribution $\mathbb{P}_Z$ ($Z \sim \mathbb{P}_Z$) in to generated artificial data X_{fake}. Z is the input of the *Generator*'s deep neural network with learning parameters θ_G, and $G(z)$ is the corresponding output. The *Generator* implicitly defines a probability distribution $\mathbb{P}_G$ as the distribution of the samples $G(z)$ obtained when $Z \sim \mathbb{P}_Z$).

$$X_{\text{fake}} = G\left(Z;\theta_G\right) \tag{1}$$

2) Discriminator

The *Discriminator* is trained to distinguish between fake data produced by the *Generator* from the real data in the historical dataset. Suppose θ_D denotes the learning parameters of the *Discriminator*. The input samples X, either real data X_{real} or fake data X_{fake}. The output is a probability p_{dis} ranging from 0 to 1, measuring to what extent the input samples belong to a real dataset.

$$p_{\text{dis}} = D\left(X;\theta_D\right) \tag{2}$$

3) Value Function $V(G, D)$

In order to complete the training, loss functions are needed to guide the updating of *Generator* and *Discriminator* parameters. According to the WGAN [28], the loss function L_G and L_D can be designed as (3) and (4).

$$L_G = -\mathbb{E}_Z\left[D\left(G\left(Z;\theta_G\right)\right)\right] \tag{3}$$

$$L_D = \mathbb{E}_Z\left[D\left(G\left(Z\right);\theta_D\right)\right] - \mathbb{E}_X\left[D\left(X_{\text{real}};\theta_G\right)\right] \tag{4}$$

In the adversarial process, the generator intends to minimize the expectation of $-D(G(\cdot))$ because a large discriminator output p_{dis} representing the sample is similar to the real data. The discriminator tries to provide a small p_{dis} for the fake data by minimizing the expectation of $D(G(\cdot))$, while giving a large p_{dis} by maximizing the expectation of $D(X)$ when the input is real data. Accordingly, the adversarial process forms a two-player min-max game with the value function $V(G, D)$.

$$\begin{aligned}&\min_G \max_D V\left(G,D\right)\\&= \mathbb{E}_Z\left[D\left(G\left(Z\right);\theta_D\right)\right] - \mathbb{E}_X\left[D\left(X_{\text{real}};\theta_G\right)\right]\end{aligned} \tag{5}$$

B. Proposed SA-GAN

The behavior of a GAN relies on the *Generator* and *Discriminator* which are essentially deep neural networks. Therefore, various machine learning models can be embedded into their constructions. The WGAN applies a CNN because of CNNs' outstanding learning behavior. However, a CNN only processes information in a local neighborhood, therefore using convolutional layers alone is computationally inefficient for modeling long-range dependencies of practice data. Therefore, a SA-GAN is further developed using the *Attention Mechanism* [29] to form a self-attention module which enables both the *Generator* and the *Discriminator* to efficiently model non-local relationships.

The SA-GAN was originally developed for image generation tasks [30]. For targeted resilient networked MGs under an SEE, the self-attention module of the SA-GAN can help the MGs to better learn the sequential features of system data in the whole SEE process. Therefore, a SA-GAN with customized architecture is proposed to generate a SEE-related dataset. The SA module and the architectures of the proposed SA-GAN are shown in Fig. 2 and Fig. 3, respectively.

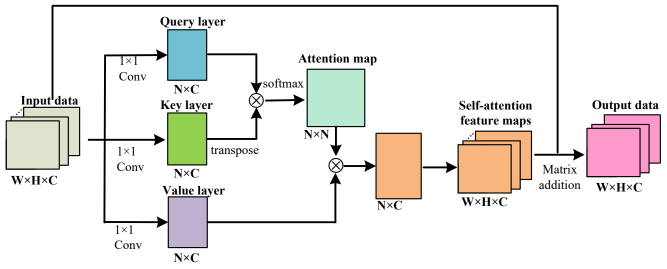


Fig. 2. Structure of self-attention module

The input data $\boldsymbol{x}$ ($\boldsymbol{x} \in R^{N\times C}$) from the previous layer are first transformed into Query, Key and Value layers to calculate the attention. Convolutions with kernel size 1 are applied to form Query ($\boldsymbol{Q} = \boldsymbol{W_f x}$), Key ($\boldsymbol{K} = \boldsymbol{W_g x}$) and Value ($\boldsymbol{V} = \boldsymbol{W_h x}$) matrixes with the shape N × C where N = W × H. Then, the matrix dot product and softmax function (6) are used to generate an attention map, with the shape N × N.

$$\begin{cases}\beta_{j,i} = \dfrac{\exp\left(S_{ij}\right)}{\sum_{i=1}^{N}\exp\left(S_{ij}\right)}\\ S_{ij} \in \boldsymbol{S} = \boldsymbol{QK}^{\mathrm{T}}\end{cases} \tag{6}$$

where $\beta_{j,i}$ indicates the extent to which the model attends to the i^{th} location when the j^{th} region is synthesized. The corresponding attention map describes each pixel's attention score on every other pixel. Then, attention weights are obtained by the matrix dot product of $\boldsymbol{V}$ and the attention map. The attention weights describe each pixel's total attention score throughout all pixels. Holding this feature, they are further reshaped to be self-attention feature maps $\boldsymbol{a}$ with shape W × H × C.

Finally, a learnable scalar γ initializing as 0 is set to multiply the output of the attention layer and add back the input feature map. The output of the self-attention module is (7).

$$\boldsymbol{y} = \gamma\boldsymbol{a} + \boldsymbol{x} \tag{7}$$

The learnable scalar γ causes the self-attention module to do nothing initially, before gradually learning to assign more weight to non-local evidence. The learnable scalar γ benefits the learning process by learning the easy task first and progressively increasing the complexity of the task [30].

Two elements are concerned in an event related changeable environment: uncertain DG output (wind power and photovoltaics (PV) in this paper), and line damage conditions. Accordingly, the historical database contains one-day (24 hours) wind $\boldsymbol{P}_{\mathbf{W}}$ and solar $\boldsymbol{P}_{\mathbf{PV}}$ DG outputs, and line damage data $\boldsymbol{D}_{\mathbf{Line}}$. Set N as the number of DGs, loads and affected lines in the event tracks, the input data [$\boldsymbol{P}_{\mathbf{W}}$; $\boldsymbol{P}_{\mathbf{PV}}$; $\boldsymbol{D}_{\mathbf{Line}}$] for the Discriminator is a matrix with size [N, 24, 1] where the first three numbers are the height, width and channels.

Taking the input size [24, 24, 1] for an example, Fig. 3 shows the data processing of the SA-GAN. The *Generator* starts with a fully connected layer (F-con) for up-sampling. Then, the first de-convolutional layer (De-cov1) has, with filters, the size [4, 4, 1, 128] where the first three numbers are the height, width, and depth of each filter and the last one is the number of filters. The filter of De-conv2 has the size [4, 4, 128, 64]. Hence, the output of De-conv2 has the size [24, 24, 64], and it goes through the SA module. The SA module remains the same size and it further provides output using a convolutional layer Conv1 with a filter sized at [4, 4, 64, 1]. Similarly, the *Discriminator* has a reversed construction with 4 convolutional (Conv) layers which have filter sizes [3, 3, 1, 16], [3, 3, 16, 32], [3, 3, 32, 64] and [3, 3, 64, 128] for Conv1 to Conv4, respectively.

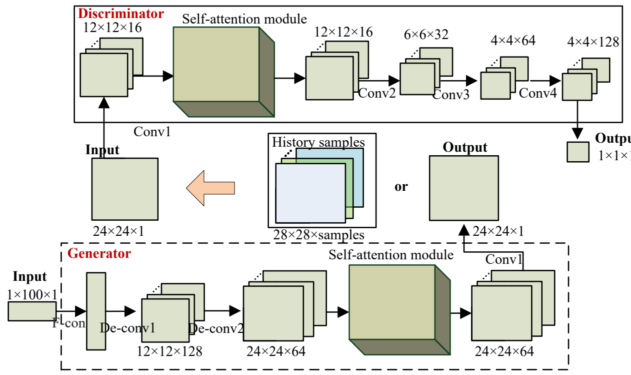


Fig. 3. Architecture of SA-GAN

To conclude, the *Generator* and *Discriminator* of the SA-GAN are designed as shown in Fig. 3. The *Generator* contains 2 De-con/(transposed convolutions) layers and 1 Conv layer to up-sample the input noise z with size [1, 100, 1] to the output $G(z)$ with the size [28, 28, 1], while the *Discriminator* includes 4 convolutional layers to down-sample the input with size [28, 28, 1] to the output p_{dis}. The self-attention modules are embedded into the de-convolutional/convolutional layers to enhance the non-local observation.

C. Training process of SA-GAN for data generation

As long as the GAN is well-trained, the *Generator* is able to capture event-related features and extend the dataset for DRL training. The adversarial training process of the SA-GAN is implemented using **Algorithm 1** which is a non-parametric unsupervised learning process without pre-defined labeling.

In the training process, the *Generator* attempts to generate fake samples with the highest possible value of $D(G(z))$ to fool the *Discriminator*, while the *Discriminator* tries to tell the difference between practical/historical samples and generated fake ones. *Generator* and *Discriminator* construction parameters update continuously during training episodes, representing a fierce competition between D and G to improve themselves. Eventually, the *Discriminator* converges to similar output values for $D(G(z))$ and $D(x)$, which means the *Discriminator* identifies the tiny difference between the real data and the fake data, and the *Generator* creates realistic event-related samples.

Training is implemented in a batch-updating style, while a gradient descent algorithm *RMSProp* with a self-adjustable learning rate is applied for weighted updates of the *Discriminator* and *Generator* neural networks. Clipping is also applied to constrain $D(x;\theta(D))$ to satisfy certain technical conditions as well as to prevent gradient explosion [28].

Algorithm 1: Training process of SA-GAN

```
Require: Sample/real DG output and line damage data from historical sample database $\boldsymbol{X}^{\mathrm{real}}$
Require: Learning rate $\alpha$, batch size M, number of iterations for discriminator $\boldsymbol{D}$ per generator $\boldsymbol{G}$ iteration $n_{\boldsymbol{D}}$, initial learning parameters for $\boldsymbol{D}$ and $\boldsymbol{G}$, $\theta^{\boldsymbol{D}}$ and $\theta^{\boldsymbol{G}}$.
While $\theta^{\boldsymbol{D}}$ has not converged do
  for $t = 0, \dots, n_{\mathrm{D}}$ do
    Sample a batch $\{X_i^{\mathrm{real}}\}_{i=1}^{\mathrm{M}} \sim \mathbb{P}_X$ from the historical data.
    Sample a batch $\{Z_i\}_{i=1}^{\mathrm{M}} \sim \mathbb{P}_Z$ from Gaussian distribution.
    $g_\theta^D \leftarrow \nabla_\theta^D\left[\frac{1}{\mathrm{M}}\sum_{i=1}^{\mathrm{M}} D\left(G\left(Z_i\right)\right) - \frac{1}{\mathrm{M}}\sum_{i=1}^{\mathrm{M}} D\left(X_i^{\mathrm{real}}\right)\right]$ (8)
    $\theta^D \leftarrow \theta^D - \alpha \cdot DRMSProp\left(\theta^D, g_\theta^D\right)$ (9)
    $\theta^D \leftarrow Clip\left(\theta^D, -c, c\right)$ (10)
  end for
  $g_\theta^G \leftarrow \nabla_\theta^G\left[-\frac{1}{\mathrm{M}}\sum_{i=1}^{\mathrm{M}} D\left(G\left(Z_i\right)\right)\right]$ (11)
  $\theta^G \leftarrow \theta^G - \alpha \cdot DRMSProp\left(\theta^G, g_\theta^G\right)$ (12)
end while
```

The completion of Algorithm 1 shows that the proposed SA-GAN is able to reasonably and credibly extend sequential DG output and line damage data. The well-trained *Generator* is extracted to generate enhanced training data for the DRL of resilient networked MGs against SEEs. The application of the SA-GAN as well as its combination with the DRL method is introduced in next section.

III. SA-GAN-DRL for Survival of Critical Loads

This section introduces the complete SA-GAN-DRL method. First, the dynamic reconfiguration of networked MGs is formulated to fit into a DDQN form. Then, a detailed reward function design and environment for networked MG reconfiguration is introduced. Finally, the entire GAN-enhanced DRL framework is concluded.

A. DDQN-based reconfiguration strategy for survival of critical loads

During the SEE process, an efficient way to reduce the impacts of events is to ensure the survival of critical loads, which is a sequential decision-making problem in a multi-step MDP. At each step, a topology configuration is determined to ensure critical loads are safely supplied under the current system power supply-demand conditions and line damages.

In the MDP, the ***state*** is the system P-Q condition and topology condition [$\boldsymbol{P}$; $\boldsymbol{Q}$; $\boldsymbol{D}_{\mathbf{topo}}$] where $\boldsymbol{D}_{\mathbf{topo}}$ is the system topology considering the last step switch status and current line damages. The ***action*** is the on/off decision of remotely controlled switches (RCSs) and the ***reward*** is the grid security condition after taking an action in a state. The DDQN method is applied to provide an adaptive reconfiguration strategy for survival of critical load during the SEE process.

The DQN method is a combination of deep neural networks and Q-learning, which updates the action-value function iteratively [31]. In this paper, a CNN is used to organize Q networks. The objective of Q-learning is to estimate the value for an optimal policy. Accordingly, the agent (operator) can decide how to optimally perform actions by learning the Q values (Q(.)). The Q network is updated with a loss function representing the mean-squared temporal difference error, as shown in (13).

$$L(\theta)=\left[r_t+\gamma\max_{A_{t+1}}\mathrm{Q}\left(S_{t+1},A_{t+1}\middle|\theta\right)-\mathrm{Q}\left(S_t,A_t\middle|\theta\right)\right]^2 \quad \gamma\in\left[0,\ 1\right] \tag{13}$$

where S_t, $\boldsymbol{A}_t$ and r_t are the state, action and reward at step t, respectively, γ is the discount factor, and θ represents the parameters which organize the Q network.

Algorithm 2: DDQN learning process

```
Initialize Q network and T-Q network with same random weights and bias. Initial replay memory. Set $D^{\mathrm{step}} = 0$. Set batch size, Episode M, step number T and Epsilon-greedy parameters.
S1: for Episode from 0 to M do
    Initialize s the environment
    for Step from 1 to T do
        Perform Epsilon-greedy method and obtain $\boldsymbol{\alpha}^{\mathrm{step}}$.
        Execute $\boldsymbol{\alpha}^{\mathrm{step}}$ in the environment and obtain the reward value.
        Organize new state $\boldsymbol{S}^{\mathrm{step}+1}$.
        Add record [$\boldsymbol{S}^{\mathrm{step}}$, $\boldsymbol{\alpha}^{\mathrm{step}+1}$, $r^{\mathrm{step}+1}$, $\boldsymbol{S}^{\mathrm{step}+1}$, $D^{\mathrm{step}}$,] in memory.
        If topology is infeasible do
            Update T-Q as Q
            Break;
        End if
        If conditions for replay are satisfied do
            Train Q network (Q-CNN)
            If Step = T do
                Update T-Q as Q
            End if
        End if
    End for
End for
S2: Obtain the Q network.
```

With two separated Q networks, the DDQN method improves on the original DQN method by decoupling the action selection and action evaluation [32]. The original Q network is used to select the action with the maximum Q value while the T-Q network evaluates the Q value of the selected action. The T-Q network is a fixed network which is not updated in the Q network updating process. The fixed feature enhances efficiency and stability in the learning process. The loss function (13) is adjusted into (14) accordingly.

$$L_t\left(\theta\right)=\begin{cases}\left[r_t-Q\left(S_t,A_t\middle|\theta\right)\right]^2 & (t=\mathrm{T})\ \text{otherwise,}\\ \left[r_t+\gamma\max\limits_{A_{t+1}}\mathrm{T}\text{-}Q\left(S_{t+1},A_{t+1}\middle|\theta^{Tar}\right)-Q\left(S_t,A_t\middle|\theta\right)\right]^2 & \end{cases} \tag{14}$$

where T is the total number of steps.

According to (14), defining a suitable reward function is an indispensable part of completing the learning process for DRL methods. For the adaptive reconfiguration of networked MGs, the action (reconfiguration strategy) should first ensure that the critical loads have access to available DGs. Further, the security constraints such as voltage and branch flow limits should be considered to ensure critical loads can be successfully supplied.

Algorithm 2 shows the learning process of the DDQN in a compact form. A detailed version of the process can be found in [1], [32].

B. Design Reward and Environment

The reward function (15) is designed to help determine the Q network for the survival of critical loads problem supported by adaptive reconfiguration.

$$r_t\left(\boldsymbol{S}_t,\boldsymbol{A}_t\right)=\begin{cases}f_{\mathrm{con}}\left(\boldsymbol{\alpha}_t\right)-\sum f_{\mathrm{AC}}\left(\boldsymbol{P}_t,\boldsymbol{Q}_t,\boldsymbol{\alpha}_t\right) & \mathrm{S}_{\mathrm{con}}=1\\ f_{\mathrm{con}}\left(\boldsymbol{\alpha}_t\right) & \mathrm{S}_{\mathrm{con}}=0\end{cases} \tag{15}$$

where $\mathrm{S}_{\mathrm{con}}$ is the signal showing whether the topology is feasible, and $f_{\mathrm{AC},i}(.)$ is the function related to AC power flow results. If $\mathrm{S}_{\mathrm{con}}$ = 1 (topology feasible), $f_{\mathrm{con}}(.)$ provides the reward w; otherwise, f_{con} (.) gives punishment –w. The feasible topology should ensure each critical load is supplied by at least one DG and there are no RCSs switched on in damaged lines. This part will be introduced in detail in the following environment building step. The variable $f_{\mathrm{AC},i}(.)$ consists of voltage violation $p_{\mathrm{vol},j}$, system power loss $p_{\mathrm{loss},i}$, branch overflow $p_{\mathrm{bran},l}$, and power unbalance $f_{\mathrm{pb}}(.)$. This is given by (16)

$$f_{\mathrm{AC}}\left(\boldsymbol{P}_t,\boldsymbol{Q}_t,\boldsymbol{\alpha}_t\right)=\sum_{j\in node}p_{\mathrm{vol},j}+p_{\mathrm{loss}}+\sum_{l\in bran}p_{\mathrm{bran},l}+f_{\mathrm{pb}}\left(\boldsymbol{P}_t,\boldsymbol{\alpha}_t\right) \tag{16a}$$

$$f_{\mathrm{pb}}\left(\boldsymbol{P}_t,\boldsymbol{\alpha}_t\right)=\sum_{j\in load}p_{\mathrm{L},j}+p_{\mathrm{loss}}-\sum_{j\in DG}p_{\mathrm{DG},j} \tag{16b}$$

Specifically, in the learning process, the penalties in the reward are divided into hard constraints and soft constraints. The hard constraints are reflected by f_{con} (.) which leads to an MDP "game over" if it is triggered ($\mathrm{S}_{\mathrm{con}}$ = 1). The soft constraints are the other penalties which reduce the reward value instead of triggering an immediate "game over".

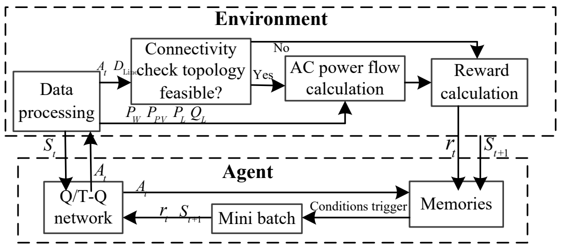


Fig. 4 Interaction of agent and environment

As shown in Fig. 4, the reward is calculated based on environmental the feedback. The environment, which contains a graph-theory-based connectivity (topology) check and an AC power flow calculation, simulates the system conditions after the Q network generated action is applied. After clearing the connectivity check module ensuring each critical load is supplied, AC power flow will be performed in a newly updated networked MG system. The process of implementing that environment is shown in **Algorithm 3**.

Algorithm 3: Environment implementation

```
Require: Select critical load set $L_{cri}$. Obtain current state $S_t$ and extract DG output $P_{DG,t}$, load amount $L_t$ and destroyed lines $D_t$ for current step. Obtain action $A_t$ from the agent.
Set k = 1. Initialize MG list as an empty set {}
If switch on destroyed lines then
    Topology infeasible
Else:
    While $i$ in DG node set do:
        If DG nodes $i$ has not been recorded then
            Search from DG nodes $i$ and label nodes connected to $i$. Record all the nodes in set $MG_k$ connected to DG nodes $i$ and extend MG list with $MG_k$.
            k = k + 1
        If all the DG nodes have been labeled:
            Break
    Check critical node set $L_{cri}$
    If any nodes in $L_{cri}$ are not recorded:
        Topology infeasible
    Else:
        Topology feasible
        Get updated MG list {$MG_1$,…$MG_{k-1}$}
If Topology feasible then
    Perform AC power flow for updated MGs in the MG list
    Record power flow result
Else:
    Game over and stop the MDP at step $t$.
Obtain the reward at step $t$ according to (15), and form $S_{t+1}$
```

C. GAN-enhanced DQN

The whole framework of the GAN enhanced DRL method is concluded in Fig. 5. First, the initial training data, which can be either historical or forecast data in a sequential Markov form, is used to perform the SA-GAN training process. As long as the generative learning process **Algorithm 1** converges, the *Generator* is obtained to generate enhanced training data for the deep RL method. The SA-GAN extracts event related features such that it can generate enough SEE-related scenarios without the limitation of scarce event data.

The DDQN method is performed using the enhanced training data. **Algorithm 2** and **Algorithm 3** are performed as the pre-training processes such that the Q-network may capture the topology and power flow related features of the networked MGs under the SEE process. Finally, the agent containing the well-trained Q network is obtained. Accordingly, the agent can give corresponding reconfiguration strategies regarding current DG output, load amount, and line damage conditions. This is called adaptive reconfiguration because the agent can flexibly adjust network configurations for the survival of critical loads based on event conditions.

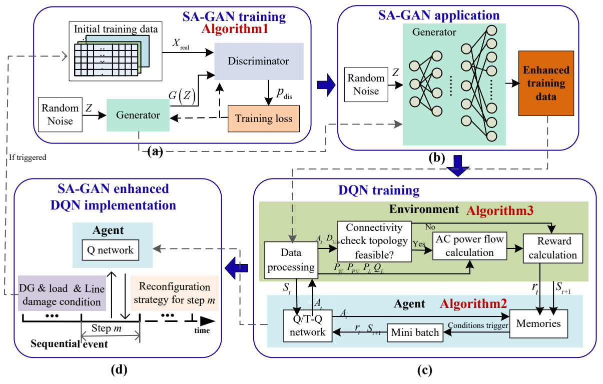


Fig. 5. The SA-GAN enhanced DQN

Note that the SA-GAN-DDQN method can have a model-free application since the well-trained agent does not need system models to provide proper actions. Moreover, the SA-GAN can be further improved by incorporating newly updated event and system information if the training conditions are trigged. Through fast-generated adaptive reconfiguration of networked MGs, the purpose of the proposed method is to ensure that each critical load is safely supplied under uncertain conditions during the SEE process. The SA-GAN-DRL method can also be used in other DRL methods

IV. Case Study

This section introduces the training and application performances of the proposed GAN enhanced DRL method. First, the data generation performances of GAN methods are shown and the improvement of the SA-GAN is presented by comparing with the original GAN [22], [26]. Then, the functions of the SA-GAN for the DRL method are demonstrated by comparing the original DDQN [1] and the SA-GAN enhanced DDQN.

Two systems are used: the two DGs integrated 7-bus system and twelve DGs integrated IEEE 123-bus system. The time horizon is set as 24 steps (24 hours), and each network reconfiguration step is set as 1 hour. The DRL codes and the corresponding environment are written and compiled in Python 3.7. Neural networks are built using TensorFlow 2.2 and Keras 2.4. Pypower 5.1 is applied to solve the power flow calculations in the environment. All simulations were conducted on a computer with Intel(R) Core (TM) i7-8550U CPU and 16 GB RAM. All the GAN related models in this paper are trained using the RmsProp optimizer with a mini-batch size of 32, while the DQN related models are trained using the Adam optimizer with a mini-batch size of 30.

A. SA-GAN based data processing

The generated data are divided into two categories. The first category is the uncertain sources: DGs (wind power and PVs) output. The second category is the hurricane track with sequentially damaged lines.

The wind and solar datasets are obtained from the NREL Wind and Solar database [33], [34]. A total of 12 wind farms and 12 solar power plants located in the north-western part of Texas are selected. One-day data with one-hour intervals is taken as an example, and 100 samples are extracted from the summer period (June - August) as the training and validating datasets.

The line damage data is generated following some sequential features of hurricanes. The sequential and regional properties [8], [11] are considered in this paper:

1) The line damages have sequential features. For instance, lines L4 and L6 are possibly damaged lines only if line L2 is destroyed. Lines L5 and L7 are possibly damaged only if L6 is destroyed. Lines will be not recovered in the considered time-horizon as long as they are destroyed.

2) Several possible paths exist. Since it is difficult to forecast the exact hurricane path, the trace is assumed to have two branches in each time interval (D1-D3). 8 steps and 6 steps are taken as a time interval for the 7-bus system and the IEEE 123-bus system, respectively. Further, the hurricane will eventually go through one path which means either line L4 or line L6 is destroyed, and either line L5 or line L7 is destroyed.

3) The probability of line damage is gradually reduced. There is an 80% chance that lines in D1 will be destroyed, while the chances that lines in D2 and D3 will be destroyed are respectively 70% and 60%, respectively. The hurricane will eventually go through one path or the other, which means either L4 or L6 and either L5 or L7 are destroyed.

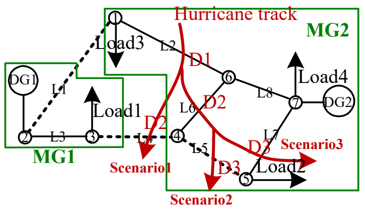


Fig. 6. Line outage conditions under hurricane

In a 24-step process, the statuses of the potential damaged lines are represented using 0 and 1 (0 for unimpacted lines and 1 for destroyed lines). Following the previous properties 1)-3), 100 samples of line outage conditions under hurricanes are generated.

In this paper, the GAN based method learns from 100 samples and generates 300 samples using a well-trained *Generator*. Note that the GAN is able to generate a database with any sizes.

1) Uncertain power supply

Both the original GAN and SA-GAN have good performance in generating uncertain power supply data. Fig. 7 and Fig. 8 compare the original historical data set and the SA-GAN-generated one. Therein, the green lines denote detailed one-day samples of wind turbine (WT)-DG output and PV output. By providing one-day samples with shapes similar to the historical ones, the SA-GAN shows the ability to capture the data feature. This is especially obvious for the PV output in Fig.7 which is near to zero at night and reaches the maximum output at noon.

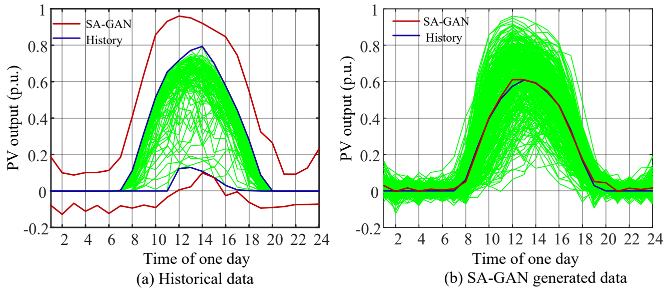


Fig. 7. PV outputs of historical and SA-GAN generated data

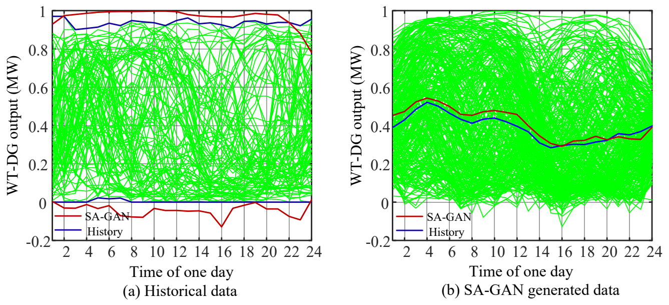


Fig. 8. WT-DG outputs of historical and SA-GAN generated data

Furthermore, the GAN-generated samples are reasonable extensions of the original data features. The red and blue curves in Fig. 7 (a) and Fig. 8 (a) respectively represent the upper and lower bounds of historical samples and SA-GAN-generated ones, while the red and blue lines in Fig. 7 (b) and Fig. 8 (b) show the mean values of historical and GAN samples, respectively. As can be observed, the upper and lower bounds of SA-GAN generated data almost contain the historical ones. Meanwhile, mean values of SA-GAN generated data and the historical data are similar. Therefore, the SA-GAN enhances the robustness of the database by considering possible extra scenarios with statistical properties similar to the original database. GAN data with more statistically similar properties can be found in [26].

2) Sequential damaged line data

For the line damage data with much stronger time-series features in a long horizon, the constructed SA-GAN has a better performance than the original GAN in [22], [26].

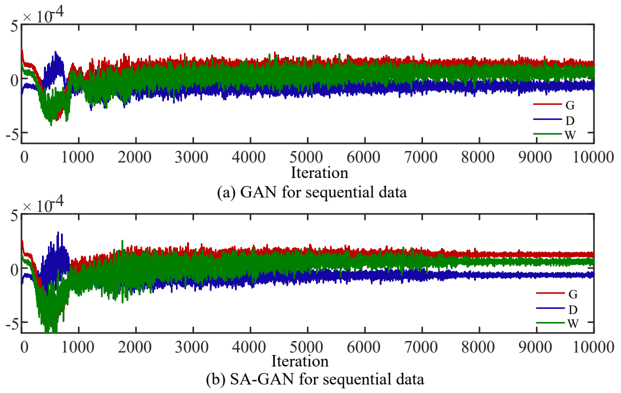


Fig. 9. Training evolution for GAN and SA-GAN on a line damage dataset

Fig. 9 shows the training processes of the GAN and SA-GAN. At start, the loss values of the *Discriminator* (blue, D) and the *Generator* (red, G) have large fluctuations as well as differences. As the training progresses, the losses of the SA-GAN in Fig. 9 (b) converge to relatively stable values with the Wasserstein distance (green, W) close to zero, while the original GAN in Fig. 9 (a) still shows large fluctuations with the same training iterations. These indicate that the SA-GAN can be successfully trained to learn underlying data features in historical datasets while the GAN fails to provide an efficient generator.

Taking the hurricane track in Fig. 6 as an example, 100 line damage samples are generated using the GAN and SA-GAN *Generators*. The no-repeat line damage conditions extracted from these samples are demonstrated in Fig. 10 and Fig. 11 with 'stars' showing the outage time. As seen in Fig. 10, four typical conditions in the training dataset were learned by the GAN, while there are six possible line damage conditions in Fig. 6. In addition, the inefficient training process of the GAN leads to some unreasonable samples. In Fig. 10, the GAN generates samples with line outages in different paths, e.g., Lines L4 and L6 as well as L4 and L7 cannot all be destroyed in one hurricane path. Compared with the GAN, the SA-GAN generates all the theoretically possible line damage conditions. Further, the typical conditions are expanded by adjusting line outage time and holding the sequential features at the same time. This enables the generated samples to enhance the robustness and adaptability of the DRL method when they are fed into the DRL training process.

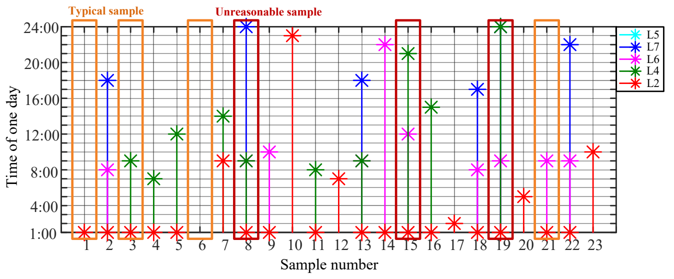


Fig. 10. GAN-generated line damage samples (no-repeat conditions in 100 samples)

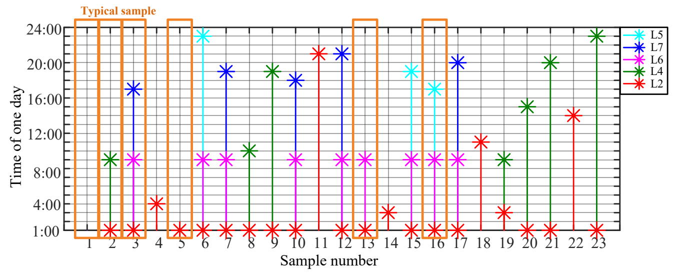


Fig. 11. SA-GAN-generated line damage samples

B. SA-GAN-DRL for survival of critical loads in the 7-bus & 123-bus systems

The SA-GAN is integrated in the DDQN method and the whole SA-GAN-DRL method is tested in the 7-bus system and IEEE 123-bus system. The historical dataset is divided into a training dataset with 80% samples and a testing dataset with 20% samples. The SA-GAN-DRL method was trained with a SA-GAN generated dataset, while the original DRL method was trained using historical training data. Test results of the SA-GAN-DRL method are compared with the DDQN method in [1].

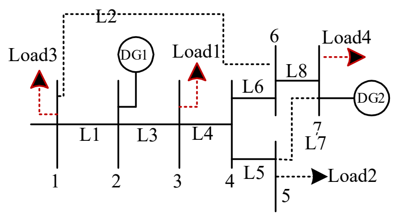


Fig. 12. The 7-bus system

Taking one hurricane path as an example, the distribution system lines will be damaged sequentially as shown in Fig. 13. The RCSs are equipped between in L1, L4 and L5. Fig. 13 presents the configurations of networked MGs over 24 h. The GAN enhanced Q network provides an adaptive reconfiguration scheme to hold critical loads. As shown in Fig. 13, L2 is damaged first. Therefore, the critical load, Load3, which is initially supplied by MG2 in Fig. 6, is transformed to MG1 in Fig. 13 (a). As the SEE evolves, switches in damaged lines are opened as shown in Fig. 13 (b) and (d). Based on DG outputs and network conditions during the whole SEE process, the GAN enhanced Q network flexibly transfers the paths of critical load supplies and avoids switches on damaged lines.

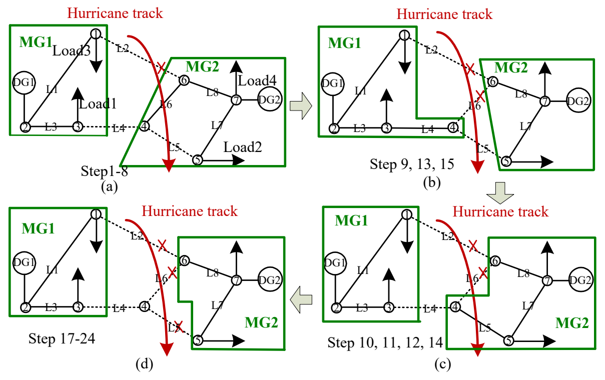


Fig. 13. Reconfiguration of networked MGs in one day

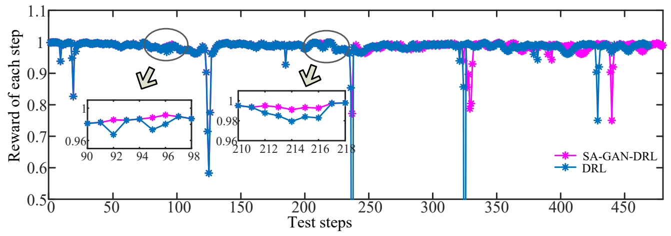


Fig. 14. Rewards of testing samples of 7-bus system

Using testing samples, the performance of the SA-GAN-DRL method (trained with GAN enhanced data) is compared with the original DRL method (trained with the initial data). Although most of the reward values are similar, there are some cases in which the SA-GAN-DRL framework (pink curve in Fig. 14) has greater reward values than the original DRL method (blue curve in Fig. 14). Since the reward is related to power balance, voltage security and power losses, a higher reward value means a better power dispatch across the whole system of networked MGs. Moreover, the proposed method ensures that all critical loads can be served with at least one available power source across the entire 480 steps (i.e., hours) of 20 samples, while the original DRL method fails to find a solution to supply some critical loads under two samples. Therefore, the SA-GAN-DRL method shows stronger robustness and better adaptability than the original DRL method according to the test results.

The performances of extending training datasets using GAN is shown in Fig. 15. Agent performances of applying different numbers (50, 100, 200, and 300) of GAN generated samples are shown using reward values during 480 test steps. As shown in the figure, the amount of critical loads shedding (with ab-normal low reward value -1) is reduced as the number of samples increases. The average reward value increases from 0.9458 to 0.9807 when the sample number increases from 50 to 300, which means a better power dispatch across the whole system using DRL decision results. Therefore, more useful training data generated by GAN can improve the application performance of DRL.

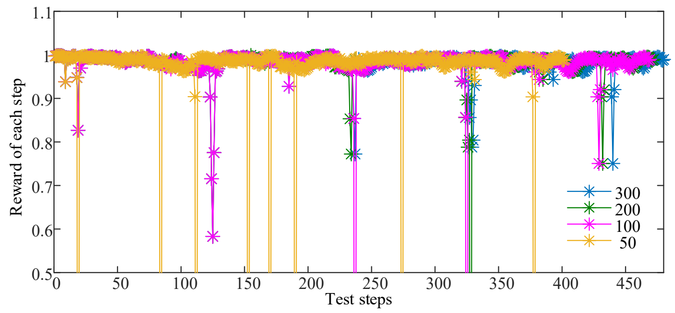


Fig. 15. Performances of training DRL with different number of GAN methods generated samples

In addition to the original dataset, 20 out of example samples with 240 steps are tested. They are generated with 20% fluctuation s from the boundaries of the source database. The performance of DDQN and SA-GAN-DDQN are shown in Fig. 16 below. Although the tested scenarios are not included in the original dataset, DDQN successfully provided reasonable actions in 156 of 240 steps, while SA-GAN-DDQN improved the number of reasonable actions to 226 steps. In other words, the DDQN itself can provide some adaptivity for some out of example cases, but the ability is limited. Using our proposed GAN based dataset extension, the adaptivity of DDQN can be further enhanced to satisfy more out of example cases.

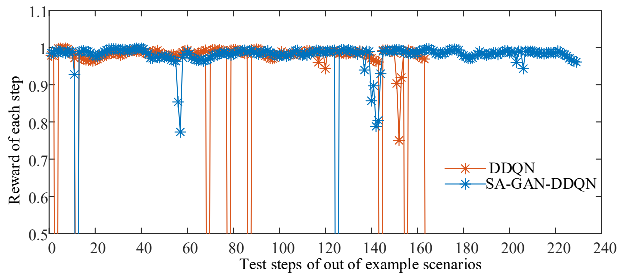


Fig. 16. Performances of SA-GAN-DRL in scenarios not included in the source database

Fig. 13 shows the MG formation changes under a SEE. The mutual assistance among MGs is more obvious in the large-scale IEEE 123-bus system, which is divided into five networked MGs as shown in Fig. 17. Figs. 18 (a) and (b) show the training process of the SA-GAN-DDQN in the IEEE 123-bus system. 16 actions are designed for the network adjustment of networked MGs. The number of steps for each episode with successful survival of critical loads is shown in Fig. 18 (a). 24-hour load survival means a successful adaptive resilient defense of networked MGs, while any incorrect switch actions (lost critical loads or switches on damaged lines) in the process directly lead to the end of an episode. Well-trained Q networks are obtained after 850 episodes. The relatively stable 24-step load surviving at the end of the training process demonstrates that the SA-GAN-DRL method successfully learned how to form feasible topologies for the purposes of survival of critical load by providing on/off switch decisions in the dynamic SEE process.

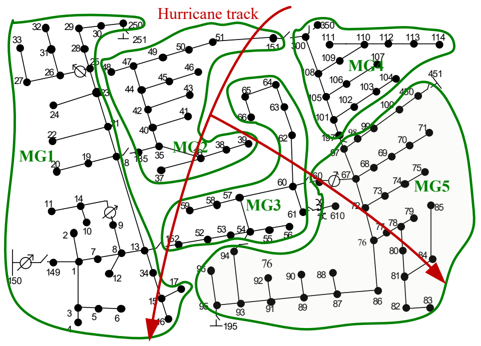


Fig. 17. Reconfiguration of networked MGs in one day

The training result of the IEEE 123-bus system is tested to show the benefit of the proposed method. In the 480 test steps in Fig. 19, the SA-GAN-DRL method, with an average reward value of 1.93, realizes more effective energy dispatch than the DRL method, which has an average reward value 1.90. That is because the two methods have different learning results. Under the same SEE condition (DG outputs and line damage conditions), the resilient defenses of the DRL method and GAN-DRL method are shown in Fig. 20.

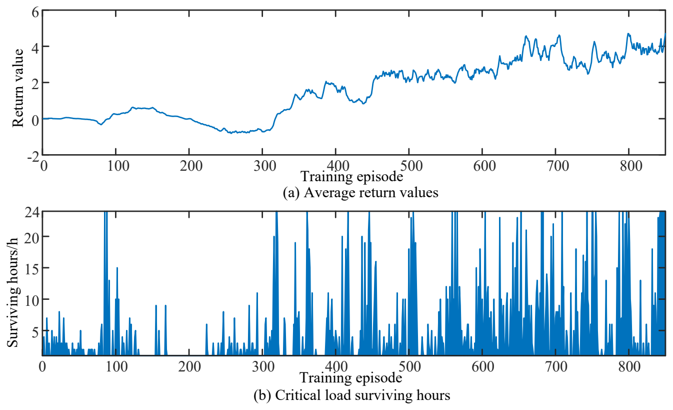


Fig. 18. Training process

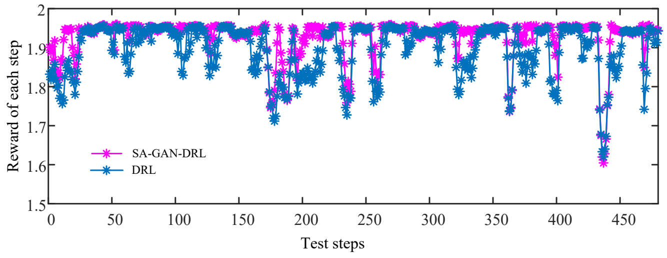


Fig. 19. Rewards of testing samples of the IEEE 123-bus system

In Fig. 20, the original system in Fig. 17 is transformed into a compact form by forming load and DG blocks. Therein, red nodes contain DGs, green nodes contain loads, and the black nodes contain buses without DGs or loads. Specifically, green nodes with red numbers represent critical loads. With line damages shown in the figure, both methods have learned to hold critical loads through mutual assistance among MG3, MG4 and MG5. However, the GAN-DRL scheme obtains higher reward value by connecting MG1 and MG2. Although this action does not benefit the survival of critical load, it reduces power losses and improves energy consuming efficiency.

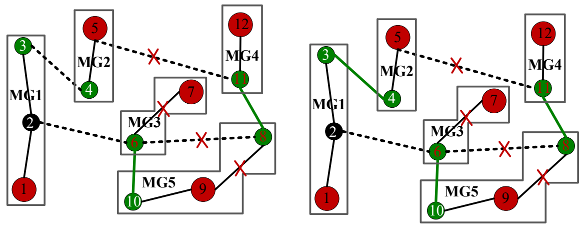


Fig. 20. Resilient responses of GAN-DRL and DRL

Overall, the SA-GAN-DRL method improves the adaptability of the original DRL method in two aspects:

- It provides more stable feasible actions. As shown in Fig. 14, when the action of the DRL method cannot form a feasible topology with a reward value of -1, the SA-GAN-DRL method gives feasible actions.
- It provides actions with better performance. As shown in Fig. 19, when the actions of both methods are feasible, the SA-GAN-DRL actions normally have higher reward values.

The proposed SA-GAN-DRL method is compared with optimization-based methods. A widely used mixed-integer linear programming (MILP) model [4] and an accurate mixed-integer second-order conic programming (MISOCP) model [35] are employed. To compare the performance of different methods, the objective value is set as the supplied load amount minus power losses in the entire system. Fig. 21 shows the objective values from different methods during a complete 24 steps Markov process. The process is realized by recursively applying MILP and MISOCP models in each step. The data-driven MILP may lead to better performance and higher accuracy. As listed in Table I, the MILP model has satisfactory computational speed, however, the result has a large optimization gap because of the relatively rough relaxation of power flow as well as power losses. The accurate MISOCP model provides the highest objective value, however, the computation time is much longer than the other two methods. If compared with the optimization methods, our proposed method has extremely fast computation time of 0.091 s and acceptable optimization gap of only 1.81%.

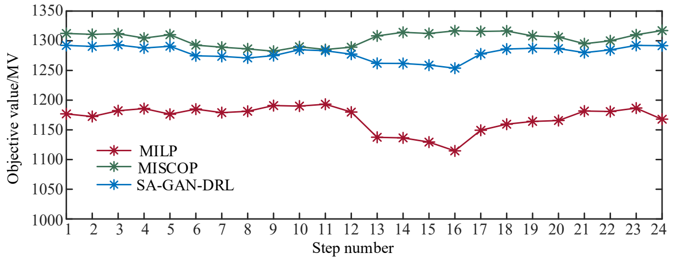


Fig. 21. Rewards of testing samples of the IEEE 123-bus system

TABLE I
COMPARISONS WITH OPTIMIZATION-BASED METHODS

Method	Computation time (s)	Objective value (MW)	Optimization gap
MILP	3.57	1169.46	10.29%
MISOCP	778.86	1303.53	--
SA-GAN-DRL	0.09	1279.91	1.81%

V. Conclusion

The changeable environment and power system conditions during the SEE process require high adaptability of the resilient response scheme. To improve grid resilience during the SEE process, this paper proposes a DRL method enhanced by the SA-GAN to ensure survival of critical load using networked MGs. The GAN is developed with the *Attention Mechanism* to better learn the sequential features of system data in the entire SEE process. Further, the DRL method is equipped with the SA-GAN to implement efficient learning with limited data. According to the case studies, the proposed SA-GAN reasonably extends the dataset by generating new and distinct extra data that captures the intrinsic features of the original data, and the proposed SA-GAN-DRL method provides robust and adaptive scheme for the survival of critical load during SEE processes.

REFERENCES

[1] J. Zhao, F. Li, S. Mukherjee, and C. Sticht, "Deep reinforcement learning based model-free on-line dynamic multi-microgrid formation to enhance resilience," *IEEE Tran. on Smart Grid*, vol. 13, no. 4, pp. 2557-2567, July 2022.

[2] J. Wang and H. Gharavi, "Power grid resilience," *Proc. IEEE*, vol. 105, no. 7, pp. 1199-1201, 2017.

[3] Z. Li, M. Shahidehpour, F. Aminifar, A. Alabdulwahab and Y. Al-Turki, "Networked microgrids for enhancing the power system resilience," *Proc. IEEE*, vol. 105, no. 7, pp. 1289-1310, 2017.

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

[5] L. Che and M. Shahidehpour, "Adaptive formation of microgrids with mobile emergency resources for critical service restoration in extreme conditions," *IEEE Trans. on Power Syst.*, vol. 34, no. 1, pp. 742-753, 2019.

[6] Q. Shi, F. Li, M. Olama, et al "Network reconfiguration and distributed energy resource scheduling for improved distribution system resilience," *Int. J. Electr. Power Energy Syst.*, vol. 124, Jan. 2021.

[7] J. Zhao, Q. Wu, N. D. Hatziargyriou, F. Li and F. Teng, "Decentralized data-driven load restoration in coupled transmission and distribution system with wind power," *IEEE Trans. on Power Syst.*, vol. 36, no. 5, pp. 4435-4444, Sept. 2021.

[8] R. P. Liu, S. Lei, C. Peng, W. Sun and Y. Hou, "Data-based resilience enhancement strategies for electric-gas systems against sequential extreme weather events," *IEEE Trans. on Smart Grid*, vol. 11, no. 6, pp. 5383-5395, Nov. 2020.

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

[10] C. Wang, Y. Hou, F. Qiu, S. Lei, and K. Liu, "Resilience enhancement with sequentially proactive operation strategies," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 2847–2857, Jul. 2017.

[11] W. Yuan, J. Wang, F. Qiu, C. Kang, and B. Zeng, "Robust optimization based resilient distribution network planning against natural disasters," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2817–2826, Nov. 2016.

[12] F. Li and Y. Du, "From AlphaGo to power system AI," *IEEE Power Energy Mag*., vol. 16, issue 2, pp. 76-84, Mar. 2018.

[13] Y. Du, F. Li, J. Li, and T. Zheng, "Achieving 100x acceleration for N-1 contingency screening with uncertain scenarios using deep convolutional neural network," *IEEE Trans. Power Syst.*, vol. 34, pp. 3303-3305, 2019.

[14] J. Zhao, F. F. Li, X. Chen and Q. Wu, "Deep learning based model-free robust load restoration to enhance bulk system resilience with wind power penetration," *IEEE Trans. Power Syst.*, early access.

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

[16] C. Chen *et al.*, "Model-free emergency frequency control based on reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 3203, no. c, pp. 1–1, 2020.

[17] Y. Zhang, F. Qiu, T. Hong, Z. Wang and F. Li, "Hybrid imitation learning for real-time service restoration in resilient distribution systems," *IEEE Trans. on Indus. Info*., early access.

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

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

[20] X. Kou, Y. Du, F. Li, H. Pulgar-Painemal, et al, "Model-based and data-driven HVAC control strategies for residential demand response," IEEE Open Access Journal of Power and Energy, vol. 8, pp. 186-197, 2021.

[21] R. S. Sutton and A. G. Barto A G., "Reinforcement learning: An introduction," MIT press, 2018.

[22] Y. Yuan, K. Dehghanpour, F. Bu and Z. Wang, "Outage detection in partially observable distribution systems using smart meters and generative adversarial networks," *IEEE Trans. on Smart Grid*, vol. 11, no. 6, pp. 5418-5430, 2020.

[23] Q. Zhang, F. Li, Q. Shi, K. Tomsovic, J. Sun and L. Ren, "Profit-oriented false data injection on electricity market: reviews, analyses, and insights," *IEEE Trans. on Indus. Info*., vol. 17, no. 9, pp. 5876-5886, 2021

[24] X. Fu. "Statistical machine learning model for capacitor planning considering uncertainties in photovoltaic power," *Protection and Control of Modern Power Systems*, vol. 7, no. 1, pp. 51-63, 2022.

[25] I. Goodfellow et al., "Generative adversarial nets," *Proc. Adv. Neural Inform. Process. Syst.*, 2014, pp. 2672–2680.

[26] Y. Chen, Y. Wang, D. Kirschen and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Trans. on Power Syst*., vol. 33, no. 3, pp. 3265-3275, 2018.

[27] X. Zheng, B. Wang, D. Kalathil and L. Xie, "Generative adversarial networks-based synthetic PMU data creation for improved event classification," *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 68-76, 2021.

[28] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in *Proc.34th Int. Conf. Machine Learning*, 2017, pp. 1–10.

[29] Vaswani, Ashish, et al, "Attention is all you need." *Advances in neural information processing systems*. Pp. 5998-6008 2017.

[30] Zhang, Han, et al, "Self-attention generative adversarial networks," *International conference on machine learnin*g. pp. 7354-7363, PMLR, 2019.

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

[32] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *30th AAAI Conference on Artificial Intelligence*, 2016.

[33] https://www.nrel.gov/grid/wind-integration-data.html

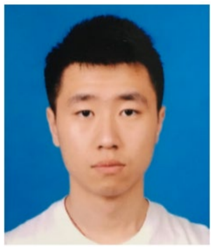
[34] https://www.nrel.gov/grid/solar-power-data.html

[35] M. Farivar and S. H. Low, "Branch flow model: relaxations and convexification - part I," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2554–2564, 2013.

Jin Zhao (S'2017, M'2020**)** received her Ph.D. degrees from Shandong University, China, in 2020 in electrical engineering. She is currently an Alexander von Humboldt Researcher hosted by TU Dortmund University, Germany. She was a Research Scientist at The University of Tennessee (UTK), USA. She was 2021 outstanding reviewers of IEEE Trans. on Power Systems and IEEE OAJPE. She is the chair of IEEE PES task force on Advanced Intelligence Techniques for Resilient Power System Restoration. Her research interests include power system resilience, transmission & distribution system restoration, renewable energy integration, resilient micro-grids and machine-learning.

Fangxing Li (S'98, M'01, SM'05, F'16) is also known as Fran Li. He received the B.S.E.E. and M.S.E.E. degrees from Southeast University, Nanjing, China, in 1994 and 1997, respectively, and the Ph.D. degree from Virginia Tech, Blacksburg, VA, USA, in 2001. Currently, he is the James W. McConnell Professor in electrical engineering and the Campus Director of CURENT at the University of Tennessee, Knoxville, TN, USA. His current research interests include resilience, artificial intelligence in power, demand response, distributed generation and microgrid, and energy markets. He was the Past Chair (2020-2021) of IEEE PES Power System Operation, Planning and Economics (PSOPE) Committee. Since 2020, he has been serving as the Editor-In-Chief of *IEEE Open Access Journal of Power and Energy (OAJPE)*.

Haoyuan Sun (S'19) is with the department of electrical engineering, University of Tennessee, Knoxville, TN, USA. He received his B.S. degree in 2019 from Xi'an Jiaotong University, Xi'an, China. His current research interests include demand response, microgrid, and power system oscillations.



Qiwei Zhang (S'17) is presently a Ph.D. student in the department of electrical engineering and computer science at The University of Tennessee, Knoxville (UTK). He received his B.S.E.E. degree from North China Electrical Power University in 2016 and M.S.E.E degree from UTK in 2018. His research interests include cybersecurity in power systems, power system optimization, and market operation.

Hang Shuai (S'17, M'19) received the B.Eng. degree from Wuhan Institute of Technology (WIT), Wuhan, China, in 2013, and the Ph.D. degree in electrical engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2019. He was also a visiting student researcher with the University of Rhode Island (URI), Kingston, RI, USA, from 2018 to 2019. He was a Postdoctoral Researcher with the University of Rhode Island from 2019 to 2020. Currently, he is a Research Scientist with the University of Tennessee, Knoxville, TN, USA. His research interests include deep reinforcement learning, microgrid optimization, bulk power system resilience.