

Learning Sequential Distribution System Restoration via Graph- Reinforcement Learning

Z. Tianqiao

To be published in "IEEE TRANSACTIONS ON POWER SYSTEMS"

April 2022

Interdisciplinary Science Department
Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Electricity Delivery and Energy Reliability (OE), Power Systems Engineering
Research and Development (OE-10)

Notice: This manuscript has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy. The publisher by accepting the manuscript for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Learning Sequential Distribution System Restoration via Graph-Reinforcement Learning

Tianqiao Zhao, *Member, IEEE*, Jianhui Wang, *Fellow, IEEE*

Abstract—A distribution service restoration algorithm as a fundamental resilient paradigm for system operators provides an optimally coordinated, resilient solution to enhance the restoration performance. The restoration problem is formulated to coordinate distribution generators and controllable switches optimally. A model-based control scheme is usually designed to solve this problem, relying on a precise model and resulting in low scalability. To tackle these limitations, this work proposes a graph-reinforcement learning framework for the restoration problem. We link the power system topology with a graph convolutional network, which captures the complex mechanism of network restoration in power networks and understands the mutual interactions among controllable devices. Latent features over graphical power networks produced by graph convolutional layers are exploited to learn the control policy for network restoration using deep reinforcement learning. The solution scalability is guaranteed by modeling distributed generators as agents in a multi-agent environment and a proper pre-training paradigm. Comparative studies on IEEE 123-node and 8500-node test systems demonstrate the performance of the proposed solution.

Index Terms—Graph reinforcement learning, distribution system restoration, distributed generation, graph convolutional networks

I. INTRODUCTION

SEVERE weather events pose increased challenges for maintaining the resilience and reliability of modern power systems [1]. When power outages occur, the distribution system restoration (DSR) is triggered to quickly restore the affected loads, leveraging advanced emerging control after the outages are isolated [2]. Researches aiming to expedite the restoration progress have been conducted to develop a self-healing and automatic restoration paradigm. Specifically, alternative network topology will be constructed through controlling smart switches, which energize these out-of-service loads by performing necessary reconfiguration while satisfying operational constraints [3], [4].

DSR is conventionally formulated by a complex combinatorial problem for network reconfiguration. This problem has been addressed mainly in three categories: 1) the heuristic or meta-heuristic algorithms [5], [6], 2) expert systems [7], and 3) mixed-integer programming (MIP)-based algorithms [8], [9]. Among these solutions, MIP-based algorithms have attracted much interest as their well-recognized optimality guarantee and formulation flexibility. Various MIP-based solutions, [8], [10]–[12] have been proposed from different perspectives to improve the restoration performance in the concept of active distribution systems (ADS). In this context, various controllable devices, such as distributed generators (DGs), microgrids and smart switches, are coordinated by an adequately designed

optimization problem centrally. It generates a restoration solution that ensures the performance and optimality of DSR. In particular, the solution is executed sequentially to control these controllable components respecting network constraints [13]. However, the existing work solves the DSR problem relying on a detailed or approximated physical power system model. Two aspects limit these model-based solutions: 1) the solution reliability can not be guaranteed if the system operators have incomplete and inaccurate network information, especially when the operator has to maintain parameters for a distribution network covering numerous nodes; 2) with the growth of the number of controllable components, the computational time of these algorithms increases rapidly, and therefore, their solutions usually have less scalability and online capability. Although a decentralized restoration scheme has been investigated in [14] for achieving better scalability, it still relies on an accurate system model.

Recent advances in deep reinforcement learning (DRL) [15] reveal the potential in solving complex decision-making problems. In a general DRL setup, the Markov decision process (MDP) is adopted to formulate the decision-making problem, and this problem is then solved iteratively by DRL-based algorithms [15]. Several DRL-based frameworks have been architected in the application of energy management systems [16], [17]. A few studies have been conducted to address the DSR problem based on DRL, showing the success in this area [18], [19]. However, they are still in the early stage, and there are still two vital challenges in solving the DSR problem which have not yet been fully addressed. One challenge is that numerous parameters, feasible decisions, and reconfiguration actions increase significantly with the size of distribution systems. Therefore, it is costly and time-consuming, considering one agent to collect central information. In [20], the authors developed a multi-agent (MA)-based safe policy learning framework for power management of networked microgrids, where the gradient information of operational limits is used to ensure safe decisions. However, as this method does not consider agents' interactions, lack of interpretability of such power system models in learning-based solutions may exist [21]. Note that the impacts from neighboring agents are different and changing after agents take different actions, and to learn cooperative behavior better, it is needed to incorporate their interactions during training. Especially in the DSR problem, agents in DRL need to make decisions based on a complete knowledge of the system structure and conditions, by which they can understand the underlying factors that affect their decision-making. Consequently, the above challenges make it difficult to apply DRL-based algorithms to DSR problems directly.

In this work, we establish a general DRL framework equipped with graph convolutional networks (GCN) called

T. Zhao is with Brookhaven National Laboratory, Upton, NY 11973, USA (e-mail:tzhao1@bnl.gov).

J. Wang is with the Department of Electrical and Computer Engineering, Southern Methodist University, Dallas, TX 75275 USA (e-mail:jjianhui@smu.edu)

Graph-Reinforcement Learning (G-RL) to address the challenges mentioned above. GCN takes the feature matrix of nodes and edges represented by a graph as the input and produces a node-level feature matrix [22]. It draws increasing attention to solve problems over graphs [23]. Note that the DSR problem is profoundly associated with the power system topology, e.g., the on/off operation of switches will affect the system topology and the energization path constructed by DGs. This work employs multi-head attention [24] as the convolution kernel to extract the relation representation between a node and its neighboring nodes in a distribution network. In sequence, we embed GCN to DRL which takes advantage of its representation ability so as to encode the relation of a graphical power system model. These graphical features are taken as the input to an RL module to guide the DG decision-making. Since multiple black-start DGs could be in a power system and energize loads, a environment [25] models the DG's interaction, where valuable information is extracted to address the DSR problem cooperatively. The G-RL-based DSR framework includes the following capabilities: 1) it can be expanded to various dynamic DSR problems using either DGs or microgrids with sequential control actions; 2) it is not limited to a typical power system model; 3) it has the scalability to address DSR in a large-scale system. The contributions of this paper are as follows:

- A novel G-RL framework is proposed to learn effective control policies for DSR problems sequentially without the knowledge of power system parameters, leveraging the graphical power system model and model-free feature of RL.
- It embeds the power system topology to the structure of GCN. Hence, DGs are guided to learn the complex restoration process via abstracting the mutual interaction and extracting latent graphical features.
- A multi-agent system (MAS) models DGs as agents, and DGs share weights among neighboring agents, making it easy to be applied to a large-scale system.
- Comparative studies are conducted to illustrate the outstanding performance of our solution, where the scalability is verified further by the IEEE 8500-node test system.

II. PRELIMINARIES AND PROBLEM FORMULATION

This section introduces preliminaries and the problem formulation to facilitate the solution design. Tables I first defines parameters and variables used in this paper.

A. Multi-agent Distribution System Restoration

Conventionally, the DSR problem is hard to solve due to its combinatorial nature [26]. Although the MILP framework [27] shows advances in the reduction of computational complexity, the increasing computational time and model-dependent structure are still not fully addressed, in particular for the distribution systems with various DGs and microgrids.

This work formulates the restoration problem by a routing model following the MAS framework. Specifically, each black-start DG is an energization depot with communication capability. Its objective is to pick up as many loads as it can subject to network constraints (e.g., voltage and power flow limits) and physical constraints (e.g., capacities and generation

TABLE I
VARIABLES AND PARAMETERS

s, \mathcal{S}	state, the space of states
a, \mathcal{A}	action, the space of actions
o	observation
R	reward function
γ	discount factor
T	time horizon
Δt	time duration
P_{hi}^{BR}	active power flow through branch (h, i)
Q_{hi}^{BR}	reactive power flow through branch (h, i)
P_i^L	active load demand of load i
Q_i^L	reactive load demand of load i
U_n	Squared voltage magnitude of node n
\bar{P}_g^{\max}	Maximum active power capacity of DG g
\bar{Q}_g^{\max}	Maximum reactive power capacity of DG g
$P_{g,t}^{\max}$	Maximum active power capacity of DG g at time t
$Q_{g,t}^{\max}$	Maximum reactive power capacity of DG g at time t
P_{hi}^{\max}	Maximum active power capacity of branch (h, i)
Q_{hi}^{\max}	Maximum reactive power capacity of branch (h, i)
U^{\max}/U^{\min}	Maximum/minimum squared nodal voltage
$r_{hi} + jx_{hi}$	Impedance of branch (h, i)
θ	learnable parameter for the Q network
S	the number of samples
N	the number of energization agents
σ	ReLU function
W	learnable weight matrix
b	bias vector
h	hidden feature
d_K	dimensionality of K_i
H	the number of attention heads
\mathcal{B}	the set of branches
\mathcal{N}^n	the set of nodes
\mathcal{N}^i	the set of neighbors to agent i
\mathcal{N}_t^L	the set of loads restored at time t

limits). Inspired by [13], black-start DGs utilize the concept of energization agents to travel along with the system through the "energization path" to energize lines and loads where they "visit". Here, the concept of node blocks introduced in [8] is adopted to facilitate the G-RL design in Section III. With this concept, nodes interconnected by non-switchable lines will be grouped as a node cell and energized simultaneously if any energization agent visits. Thus, the searching space is significantly reduced.

An illustrative example is shown in Fig. 1, where three black-start DGs are labeled in red, brown, and green nodes with a circle. They are the energization depots with a constant power capacity. Each DG is connected to a certain number of neighboring DGs, and the communication network is stationary. An energization agent departs from the depot and travels along the energization path. Agents can not go through the same energization path and can only visit one node cell. At each time-step, an energization agent determines the next energization destination to energize the system based on its own and neighboring information. This work assumes the communication network is independent of the physical

network and connected. Agents in the communication network at least have one neighboring agent and can interact with their neighbors. Therefore, the communications would not much affect the results as long as the communication network is connected and agents can receive and share information.

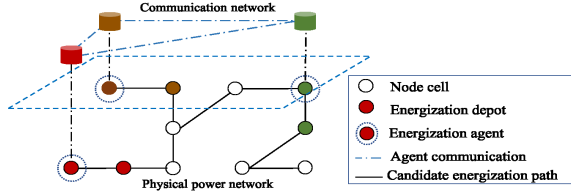


Fig. 1. Multi-agent restoration problem

B. RL for Distribution System Restoration

Here, we reformulate the proposed formulation as an MDP and establish the fundamental RL problem for system restoration. We begin by introducing the notations used to reformulate the restoration problem in RL. Let $\mathcal{P}(s' | s, a), \forall s', s \in \mathcal{S}, \forall a \in \mathcal{A}$ be a transition probability. Then, an MDP is defined by $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}(s' | s, a), R, \gamma, T)$ [28]. In our settings, each energization agent has the following pair of state and action:

1) *State* $s_{i,t} \in \mathcal{S}$: Each agent has its observation as the local state. It includes three aspects: i) Its local attributes, i.e., the current location, all previously visited node cells, and the current DG capacity after energizing the visited cells; ii) the attributes of energization paths correlated to its current location, i.e., switch status; and iii) the loading condition of the candidate node cells connected by the energization paths

2) *Action* $a_{i,t} \in \mathcal{A}$: At time-step t , an energization agent chooses an action $a_{i,t}$ from its candidate action space \mathcal{A} . In the formulated restoration problem, it is the choice of the next visiting node cell. In power networks, it indicates the switch-operation of one candidate neighboring branch.

3) *Transition probability*: \mathcal{P} : Given s_t and agents' joint actions a_t at time-step t , the system appears at the state s_{t+1} at time-step $t+1$ according to the transition probability $\mathcal{P}(s_{t+1} | s_t, a_t)$.

4) *Reward* $R_{i,t}(s_t, a_t)$: The proposed formulation covers various DSR problems, such as the sequential restoration model [8], [13] and critical load restoration [29], [30]. Given a specific restoration problem, the reward function $R_{i,t}(s_t, a_t)$ is defined as its corresponding objective such as maximizing the overall load supplied or minimizing the outage time of critical load supply, which is incurred from taking action $a_{i,t}$ at state $s_{i,t}$. In our case, the objective is to maximize the number of restored loads, and the reward is defined as

$$R_{i,t}(s_t, a_t) = P_{l,t}^L \times \Delta t \quad (1)$$

where $P_{l,t}^L$ is the restored active power of node cell l after taking action $a_{i,t}$. The following penalties are added to the reward function to ensure the feasibility and radial topology:

$R_{i,t}^l(s_t, a_t)$: A negative value equals the load demand of the node cell if an agent attempts to enter a node cell that has been already visited; otherwise, it is 0.

R_p : A negative value equals the summation of all the violated power flow constraints:

$$\begin{aligned} & -w_p \sum (max(0, P_{ij}^{BR} - P_{ij}^{\max}) + max(0, -P_{ij}^{\max} - P_{ij}^{BR})) \\ & - \sum (max(0, Q_{ij}^{BR} - Q_{ij}^{\max}) + max(0, -Q_{ij}^{\max} - Q_{ij}^{BR})) \\ & - \sum (max(0, U_i - U^{\max}) + max(0, U^{\min} - U_i)) \end{aligned} \quad (2)$$

where w_p , w_q and w_v are weights for each factor, and they are 2, 2, and 10 in our study, respectively.

R_c : A large negative value that penalizes loop formation. Since some actions may form loops but have relatively more load restoration leading to a higher total reward, this penalty should be at least greater than or equal to the maximum demand of node cells to ensure these actions are penalized. In our case study, we set this value based on the value of the system load demand, e.g., twice as great as the total demand, to penalize these actions significantly.

Remark 2.1: This work does not decide the power generation directly, which, however, is included in the observation of agents. By doing so, at each step, agents choose the next visiting node cells based on their current power capacity and other observations. If an agent decides where to visit next, all loads within this node cell will be restored simultaneously. When an agent restores a node cell, after taking the determined action, its capacity is updated by

$$P_{i,t}^{\max} = \bar{P}_i^{\max} - \sum_{l \in \mathcal{N}_{t-1}^L} \sum_{k=1}^{t-1} P_{l,k}^L \quad (3)$$

$$Q_{i,t}^{\max} = \bar{Q}_i^{\max} - \sum_{l \in \mathcal{N}_{t-1}^L} \sum_{k=1}^{t-1} Q_{l,k}^L \quad (4)$$

where $\sum_{l \in \mathcal{N}_{t-1}^L} \sum_{k=1}^{t-1} P_{l,k}^L$ and $\sum_{l \in \mathcal{N}_{t-1}^L} \sum_{k=1}^{t-1} Q_{l,k}^L$ denote all the restored active and reactive power of load demand previously. Note that if the remaining power capacity is less than the visited node cell, only load demand that equals this power capacity will be restored, resulting in a decreased reward.

Denote a policy as $\Pi_{\theta}(a_t | s_t)$, which is a distribution over candidate node cells given a set of visited and current locations. The policy returns a probability distribution over the next candidates who have not been visited with a set of visited node cells. In our case, the policy is represented by neural networks (NN). The network will return a policy respecting a specific objective and distribution network requirement by maximizing the reward. In this work, the parameterized action-value function $Q_i(\theta_k)$ for each agent at iteration k approximates the total reward by minimizing the following loss,

$$\mathcal{L} = \mathbb{E}[(R_{i,t} + \gamma \max_{a'} Q(s'_{i,t}, a'_{i,t} : \theta_{k-1}) - Q(s'_{i,t}, a'_{i,t} : \theta_k))^2] \quad (5)$$

where $s'_{i,t}$ and $a'_{i,t}$ denote the next state and action, and γ is the discount factor. The following section introduces a learning-based restoration framework which associates the structure of power networks with the structured NN and adopts DRL to address the formulated problem.

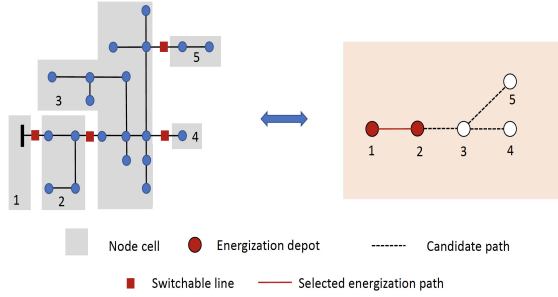


Fig. 2. Graph representation of power networks

III. TECHNICAL METHODS

To design a learning-based approach, the following aspects are considered: i) the operation condition is physically correlated in the power network topology; ii) the restoration process is closely related to the graphical structure of power systems. From these perspectives, this section constructs a model that embeds the physical power network topology to DRL.

A. Learning over Power Network Graph

We first construct the formulated problem over a graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where each node $n \in \mathcal{V}$ is the node cell, and there exists an edge $e \in \mathcal{E}$ between two nodes representing the energization path. These energization agents start from the energization depot in our settings and go along the graph to fulfill the designed objective. Here, we represent the status change of switchable lines by the selection of the next traveling node in the graph. As a simple example Fig. 2, if the energization agent determines to travel from Node 1 to Node 2, it will go through the edge e_{12} , and the corresponding switchable line connected between Node Cell 1 and Node Cell 2 will be switched on.

We will then embed this power network graph in the learning process to capture the graph dependence and correlation through graph neural networks (GNN).

B. Learning Graph Convolutional Restoration

This work utilizes the GCN model due to its inherent formulation that captures a structured graph mechanism over a power system network. However, limited attention has been devoted to generalizing the existing learning models to the structured datasets in power system applications. Recent studies construct several GNN frameworks to extract local-connected features from graphs, aiming to learn a feature representation on graphs [22]. The GCN model feeds the feature matrix integrating each node's attributes and outputs as a node-level feature matrix. The convolution operation on graphs is similar to those in convolutional neural networks (CNN). However, the feature maps in GCN are produced by the kernels convoluted across local regions in graphs. As in Fig. 3, the neighboring information in the graph domain is weighted, summed, and fed to GCN. Therefore, GCN can better capture relationships in power system graphs with the structured graph layers.

The environment is considered to be partially observable, where at time-step t , each energization agent i gets a local observation $o_{i,t}$ which is the attribute of predefined states in the graph, takes an action $a_{i,t}$, and receives a reward $R_{i,t}$. This environment is considered to be static, i.e., the operation

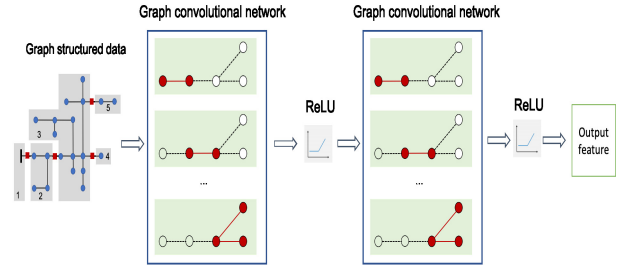


Fig. 3. Graph convolutional network

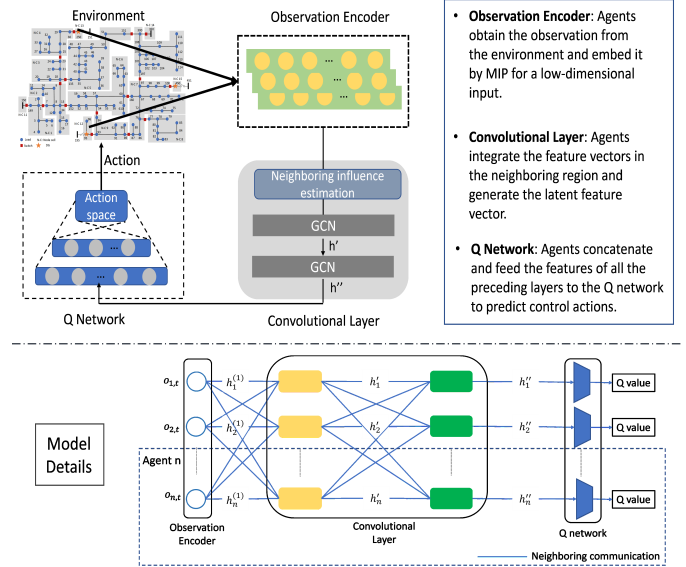


Fig. 4. The proposed learning framework

conditions (e.g., load demand, power capacity and critical load location) remain the same throughout an episode. However, across different episodes, these conditions can take random values/locations in the environment.

The proposed G-RL contains three types of modules in sequence: the observation encoding layer, neighborhood co-operation layers, and Q networks, as illustrated in Fig. 4. We assume each agent can interact with its neighboring agents to share information and learn cooperative behavior. Note that sharing the observation could be complicated due to its high dimension. To overcome this challenge, as at the bottom of Fig. 4, each agent first collects and encodes its observations by Observation Encoder to produce a low-dimensional feature vector by a layer of Multi-Layer Perceptron (MLP). Then, the feature vector is shared with its neighboring agents, and therefore, the input of Convolutional Layer contains only the features of the local and neighboring agents. With this input information, each agent produces a latent feature following the steps in Section III.B – 2). For each agent, the input of the Q network is the concatenated outputs from all the preceding layers, and we train the model based on deep Q learning given in Algorithm 1.

1) *Observation Encoder*: The observation $o_{i,t}$ includes i) the current location, all previously visited node cells, and the current DG capacity after energizing the visited cells, ii) the switch status of energization paths correlated to its current location, and iii) the load condition of the candidate node cells

connected by the energization paths. As discussed in Section III.B, each agent will share its information with its neighbors to predict coordinated actions. Sharing its local observation as a whole is complicated, and therefore, we introduce Observation Encoder to facilitate information sharing. A layer of MLP embeds $o_{i,t}$ to an m -dimensional latent space,

$$h_i^1 = f_{W_o, b_o}(o_{i,t}) \quad (6)$$

where f_{W_o, b_o} is a learnable network, with the ReLU function being the activation function. The objective of f_{W_o, b_o} is to encode the observation into a low-dimensional space. During training, it takes the observation $o_{i,t}$ as input and outputs a hidden feature vector h_i^1 . The feature vector h_i^1 efficiently encapsulates complex observation information in a low dimension and represents the agent's understanding of its observations.

2) *Convolutional Layer*: A key aspect of the coordination among energization agents is communication, especially in the MARL environment [28]. This work leverages the attention mechanism widely adopted to enhance accuracy [23]. It represents and models the neighborhood influence by the aggregation of the agent's neighboring environment information.

After computing the observation encoding h_i^1 , each agent performs the following calculations to obtain a query Γ , a key K and a value V ,

$$\Gamma_i = W_\Gamma h_i^1, \quad K_i = W_K h_i^1, \quad V_i = W_V h_i^1. \quad (7)$$

Each agent then shares the query-value pair (Γ_i, V_i) with its neighborhood agents $j \in \mathcal{N}_i$, where we also include agent i in \mathcal{N}_i to improve the awareness of attention on its condition. After receiving this pair from neighbors, we embed the value representation of the local region in GCN. Inspired by the attention mechanism [23], we first calculate a similarity score for the key,

$$\Omega_{ij} = \text{softmax}\left(\frac{\Gamma_j K_i}{d_K}\right), \quad (8)$$

This score is assigned as a weight to each of the incoming values, and we compute a weighted sum of the incoming values followed by a linear transformation, i.e.,

$$a_{ij} = \sum_{j \in \mathcal{N}_i} \Omega_{ij} V_j \quad (9)$$

where a_{ij} is the attention score that represents the neighboring importance to agent i in determining the policy.

In this work, these neighbors are pre-defined on a communication network. One can extend it by various definitions of distances such as the node cell distance and the electrical distance between DGs. Therefore, the formulation of a_{ij} can be applied to restoration problems on various network structures since it relaxes the concept of agent's neighbors.

We then combine the representation of energization agents with their importance through a transformation to set up the overall influence of neighboring conditions, i.e.,

$$h_i' = \sigma\left(W_{\text{out}} \sum_{j \in \mathcal{N}_i} a_{ij} V_j + W_0 h_i\right) \quad (10)$$

W_{out} and W_0 are the learnable weights. The weighted sum of neighboring values aggregates the information importance from the encompassing situation for obtaining efficient restoration policy. The attention mechanism expands the agent's focus based on the environment conditions (e.g., load demand and locations) to a larger scale. For example, the original emphasis of energization agent i on its neighbors is distinct due to its limited situational awareness. The attention mechanism receives the latent feature from the neighboring agents directly from the GCN layer. The messages carried by the hidden states enrich their environment knowledge since these additional messages include its adjacent agents' knowledge. As a result, this mechanism facilitates agents to determine actions with additional information.

The above single-head attention is extended to multi-head attention to increase the expressiveness further and simultaneously visit the neighborhood from other subspaces at various positions. For attention head m , the functions of a_{ij}^m and W_{ij}^m have the same formulation as (9) - (10), and the overall importance $h_i^{m'}$ is given by

$$h_i^{m'} = \sigma\left(W_{\text{out}}\left(\frac{1}{H}\left(\sum_{m=1}^H \sum_{j \in \mathcal{N}_i} a_{ij}^m v_j^h + W_0 h_i^{m'}\right)\right)\right) \quad (11)$$

Note that more attention heads give more relation representations. Besides, multiple GCN layers can extract higher-order relation representations, capture the inner-features among agents effectively, and facilitate restoration actions.

3) *Q network*: Since an energization agent takes discrete actions (i.e., the selection of the next traveling node), we deploy DQN in the RL module, but it is worth noting that other RL algorithms are also applicable to the proposed framework. As in Fig. 4, agent i concatenates all GCN layers' features and feeds them into the Q network. The value representation and features from neighboring agents are assembled and reused to learn restoration cooperatively in different ranges. The Q network determines the action with the ϵ -greedy which maximizes the Q-value and meanwhile, minimizes the following loss to optimize the policy,

$$\mathcal{L}(\theta) = \frac{1}{S} \sum_{s=1}^S \frac{1}{N} \sum_{i=1}^N \left(R_{i,t} + \gamma \max_{a'} Q(s'_{i,t}, a'_{i,t} : \theta_{k-1}) - Q(s'_{i,t}, a'_{i,t} : \theta_k) \right)^2 \quad (12)$$

where θ denotes all trainable variables. The formulations of the Q-network and (12) focus on both reward maximization and neighboring influences. In our problem, the power flow verification is only revealed after node energizations, where the agent only receives the penalty in the future. Here, we adopt the idea of n-step Q-learning and add this penalty to the final reward.

C. Pre-Training Paradigm

To extend the solution to a large-scale system, we introduce a pre-training paradigm similar to the idea of curriculum training [31]. The proposed framework does not rely on the number of energization depots. Transferring all agents'

network parameters for a more straightforward task to a complex task would be a good initialization for the complex task. This establishes a curriculum of tasks with increasing complexity [31]. In the setting of curriculum learning, agents learn better when samples are organized in a meaningful order that illustrates more complex ones gradually. The proposed framework can be applied to tasks with arbitrary numbers of agents and targets (i.e., node cells), and the agents share their parameters. According to [32], this enables us to directly use a policy from a task with N agents and M targets to a different task with N' agents and M' targets. Transferring all agents' network parameters for a more straightforward task to a complex task would be a good initialization for the complex task. This establishes a curriculum of tasks with increasing complexity [31]. Specifically, energization agents firstly learn to cooperatively restore the system in a small team with large capacities. Rather than achieving the final goal, this stage focuses on providing the agents with environmental knowledge and cooperative behaviors beforehand, by which their policies are bootstrapped to accomplish the complex objective. In other words, they share parameters with new members and apply the previous knowledge to a new scenario, and this knowledge is gradually adapting to the complex task. As proved in [33], if two MDPs have a small distance, the pre-trained Q-function from the simpler task would be closer to the optimum than the random initialization.

D. Implementation

1) *Loop detection*: For a given system, we assume the system topology is known, i.e., line connection and node locations. After agents predict the actions, we can directly connect the associated lines to obtain the topology if agents take these actions. We detect loop formation by running either a search algorithm (e.g., Depth First Search) or a python-based network analysis tool (e.g., NextworkX).

2) *Radiality guarantee*: Note that the model-free feature makes GCN challenging to handle the radial constraints directly. Although it is unlikely to take the actions that have been already significantly penalized during training, GCN still has the possibility of taking actions that violate the radial constraints. Therefore, a backup approach is introduced to avoid further forming loops during implementation. In particular, we first produce Q values for several possible actions from a given state. Then, we take the actions based on the size of Q values and check if there exists any loop. If the action with the largest Q value forms a looped topology, we will omit it and test the next action with the 2nd largest Q value, and so on.

The solution implementation can be summarized in the following steps.

- 1) The distribution network is first mapped to a graph. An adjacent matrix represents its topology. The node cell containing a black-start DG is modeled as the energization depot. Each energization agent departs from the depot to energize the system cooperatively according to the network structure.
- 2) Graph convolutional restoration is then constructed following the multi-attention mechanism in Section III.B-2). It encodes the local observation, integrates neighboring

features on the graph and generates the latent feature vector $h'_{i,t}$ to predict the Q-value. Stacking more GCN layers gradually improves an agent's observation field and cooperation scope as more information is aggregated.

- 3) After receiving the latent feature, the Q network predicts the Q-value. The Q-learning process is given in Algorithm 1. The output of the Q-network is the best next node cell to be energized. Here, we adopt fitted Q-learning [34] to learn parameters efficiently. In particular, fitted Q-learning utilizes experience replay with a random minibatch of samples rather than update the Q-network sample-by-sample.

The learning process is terminated when it either reaches the maximum learning episode or restores all loads.

Algorithm 1 Learning Q-function

Initialize: Output features from GCNs, K relayed to fitted Q-learning, L episodes and T sample size

DistEvent-VC: Learn parameter set θ of the Q-network

1: Initialize experience replay memory M to capacity N

2: **while** Training is true **do**

3: **for** Step $t = 1$ to T , agent i **do**

4:

$$e_{i,t} \leftarrow \begin{cases} \text{random } a_{i,t} \in \mathcal{A}_i, & \text{with probability } \epsilon \\ \arg \max_{\mathcal{A}_i} Q(s_{i,t}, a_{i,t} : \theta), & \text{otherwise} \end{cases} \quad (13)$$

5: Take joint action $a_t = [a_{1,t}, \dots, a_{N,t}]$ and observe the rewards $R_t = [R_{1,t}, \dots, R_{N,t}]$ and the next state $s_{t+1} = [s_{1,t+1}, \dots, s_{N,t+1}]$

6: Store $\langle s_t, a_t, R_t, s_{t+1} \rangle$ in the replay buffer

7: **for** $i = 1$ to N **do**

8: Sample a random minibatch B of K experiences from the replay buffer

9: Update θ by Adam optimizer for B

10: **end for**

11: **end for**

12: **end while**

IV. CASE STUDIES

This section conducts case studies to verify the effectiveness of the learning-based restoration framework in terms of the optimality and scalability. We start by presenting the system and algorithm configurations. Its optimality and scalability are then demonstrated and compared with benchmark algorithms.

A. Setup

1) *Distribution network configuration*: The modified IEEE 123-node [13] and IEEE 8500-node [27] test systems are adopted in this study. We follow the concept of the node block [8] to reduce the searching space. The IEEE 123-node test system contains five substations [13]. We add additional switchable lines and DGs with the black-start capability to the original test feeder and partition the system into 15 node cells. The details of the parameters of the modified system can be found in [13]. To generate load profiles for each episode, the original load demands are multiplied randomly from 0.1 to 2 p.u. to model the light and heavy loading conditions. We adopt the IEEE 8500-node system to test the scalability, where switchable lines and DGs are also added to the original system.

TABLE II
HYPER-PARAMETERS

Hyper-parameter	DQN	MARL	G-RL
τ (Softmax factor)	-	-	0.28
Numbers of neighbors	-	3	3
Convolutional layers	-	-	2
Attention heads	-	-	8
Number of MLP layers	-	-	2
Q-network	(1024,256)	(1024,256)	affine transformation
Shared			
Discount factor	0.96		
Reply buffer	2×10^5		
ϵ and decay	0.68 and 0.995		
Optimizer	Adam		
Activation	ReLU		

The system is then partitioned into 578 node cells. The target number of DGs to be controlled is 20. The load profiles for training are generated by randomly disturbing base loads.

2) *Algorithm configuration*: The configurations of the proposed G-RL algorithm and two benchmark RL algorithms (i.e., basic single-agent DQN [34] and MARL [25]) are summarized in Table II, where only CNN are utilized in the benchmark algorithms. We manually tune the hyper-parameters of different algorithms to obtain their best performance. The parameters share the same discount, batch size, buffer capacity, optimizer, ϵ and decay. Note that all reward components are normalized, i.e., in p.u., to match the weights of NN. The benchmark results are then compared with all RL algorithms under the same operational condition.

All solution algorithms are trained and tested in Python on a personal laptop with a 2.6 GHz Intel Core i7 and 16 GB of RAM. The power flow is evaluated using pandaPower [35] when each training episode is completed. If there exists any power flow violation, the corresponding penalty R_p is added to the final reward.

For better comparison, two performance indexes are adopted to evaluate the solution algorithm: 1) the mean reward and 2) the success rate (SR). In particular, we record the reward every 100 episodes. For the SR, we calculate the percentage of successful tasks in all previous 100 episodes.

B. Modified 123-node test system

Here, we illustrate the results of the proposed solution to verify its effectiveness and performance. The system topology and its corresponding graph are given in Fig. 5. In this case, 5 energization depots, including 2 substations at nodes 150 and 350, and 3 DGs at nodes 250, 450, and 95 are deployed to restore 15 node cells. For G-RL, each episode has a different loading condition and DG capacity and constant depot locations. During the training process, the weights of NN are periodically recorded and used for testing the solution performance. The results of our solution and the benchmark RL algorithms are given in Figs. 6 - 7, where the mean reward and SR are illustrated, respectively. The SR measures the percentage of episodes which achieves the objective. For all algorithms, we calculate their min-and max-values in different training runs (i.e., shaded area) and the corresponding mean value (i.e., solid line).

As learning curves for rewards and SRs are shown in Figs. 6 - 7, the proposed solution is capable of discovering

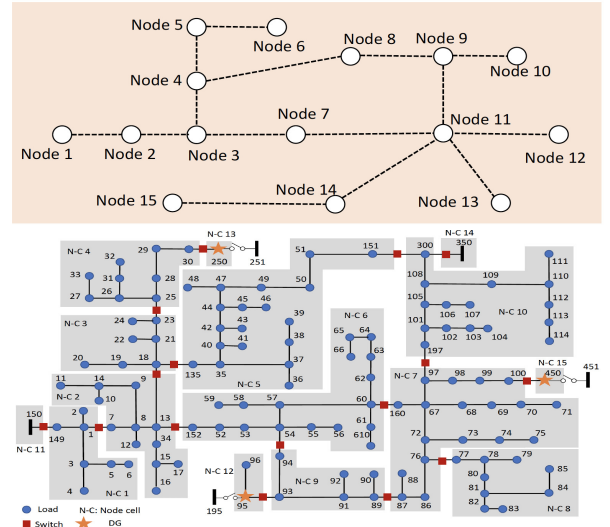


Fig. 5. Graph representation and physical topology of the modified IEEE 123-node test system

an outperforming policy. As expected, DQN has the worst performance. In contrast, our solution yields a lower loss and higher reward than the benchmark RL algorithms, and meantime, the convergence speed is faster which can facilitate us to design the solution for a large-scale system. The results of the SR comparison are given in Fig. 8. The mean SRs are 28%, 56% and 88% respectively for DQN, MARL and G-RL, which verify our solution's outstanding performance.

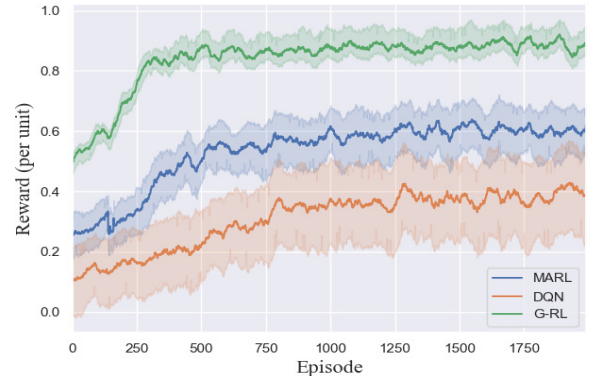


Fig. 6. Learning curves of rewards for the IEEE 123-node test system

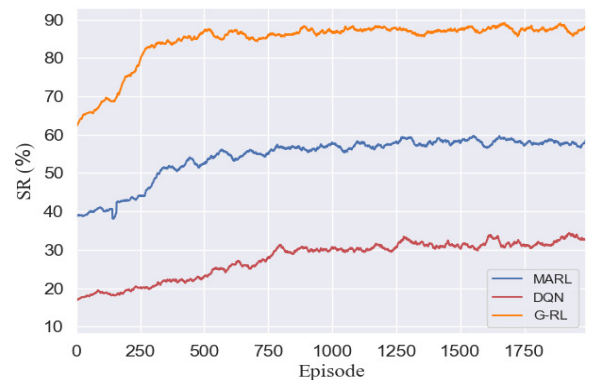


Fig. 7. Learning curves of the SR for the IEEE 123-node test system

TABLE III
TESTING RESULTS IN THE IEEE 123-NODE TEST SYSTEM

	DQN	MARL	G-RL	CPLEX
Light loading ($0.5 \cdot P_L$)	100%	100%	100%	100%
Heavy loading ($2 \cdot P_L$)	68%	92%	100%	100%
Average time (s)	1.82	1.75	1.78	35

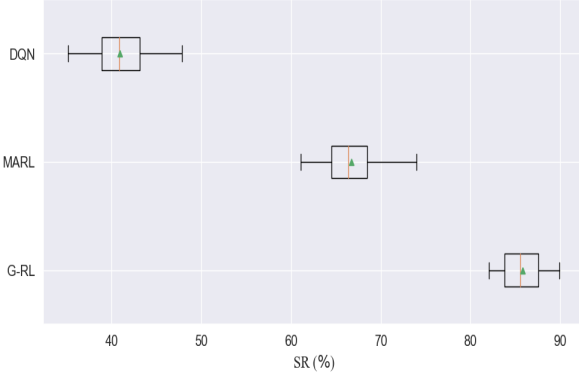


Fig. 8. Comparison of the SR for the IEEE 123-node test system

All algorithms are then evaluated by 3 test cases, and each case is unrolled with 500 episodes. To verify their efficiency, we model the problem by the fixed-time step model as in [13] and solve it by CPLEX. The restored load and computational time are given in Table III, where P_L is the system's baseline load. All the approaches return similar results as the CPLEX using less computational time when the system is in the light loading condition. However, DQN and MARL suffer a decreased accuracy, especially when the system has heavy loading, and DG capacities are marginal to the given load demand. Besides, without considering the neighboring influence, DQN has the worst performance. The message sharing helps the G-RL and MARL to discover a better policy significantly. However, MARL is unable to discover the sophisticated restoration protocol because of lacking knowledge regarding power networks. Given the same operational condition, the proposed solution has a better performance than other RL benchmarks. To better interpret the results, Fig. 9 visualizes the restoration path sequentially using the learned policy. We assume the DGs at Node 55 and Node 250 have lower power capacity. As shown in Fig. 9, energization agents are coordinated to restore the system in three steps, where 1) no loops are formulated; 2) full loads are energized, and 3) DGs energize loads based on their power capacity. For example, two node cells need to be energized in the second step. At that time, the DG at Node 450 has not enough capacity to energize the node cell next to the current energization agent's location (brown). Meanwhile, the substation at Node 150 can not energize the node cell above the current energization agent's location (red) either. If it decides to energize this cell, the system can not be fully energized since each agent can not enter a node that has already been visited. Based on the current situation, the policy coordinates the energization agents by taking actions as in the third step, restoring the full system.

C. Modified IEEE 8500-node test system

To analyze the scalability, we test the proposed solution on the modified IEEE 8500-node system. The graph is represented

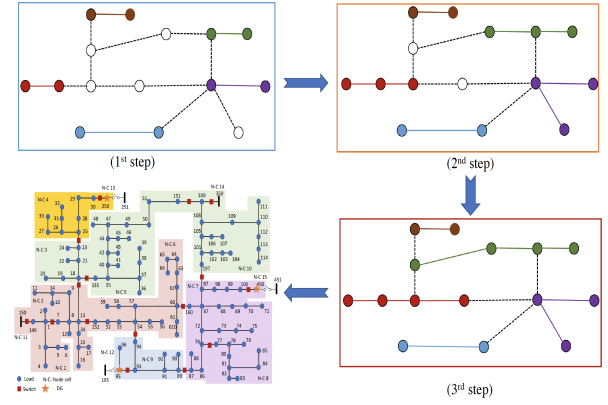


Fig. 9. Control sequences in the IEEE 123-node test system

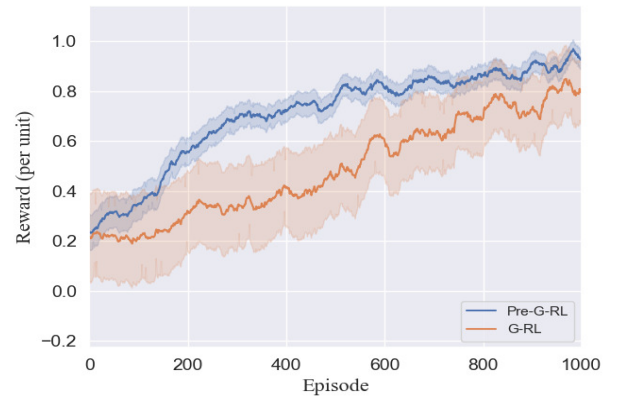


Fig. 10. Learning curves of rewards for the IEEE 8500-node test system

according to the system topology. The system contains 20 black-start DGs to energize the 578 node cells. Again, for G-RL, each episode has a random loading condition and constant depot locations. Instead of training from scratch, the proposed pre-training paradigm is deployed over different energization agents. To keep the environment being invariant, the number of neighbors to an agent is assumed to be 3. The pre-training in multi-agent systems aims to help agents learn cooperative behavior [33]. Specifically, a control policy is first pre-trained with 5 DGs. Once it is completed, the returned policy is transferred to an increased number of DGs (i.e., 10, 15 and 20) repeatedly.

The results of the solutions with/without pre-training are given in Fig. 10, where the mean rewards are illustrated for several training runs, respectively. It is shown that equipping our solution with the pre-training paradigm will significantly improve the performance of the entire learning process and is able to handle problems in a large-scale system. The mean reward, the SR, and the training time for different RL algorithms are given in Table. IV. These results verify that our solution can effectively address the restoration problem in a large-scale test system.

TABLE IV
TRAINING RESULTS OF DIFFERENT RL ALGORITHMS IN THE IEEE
8500-NODE TEST SYSTEM

	SR (%)	Mean reward (per unit)	Training time (hrs)
DQN	28	0.431	12.2
MARL	56	0.654	9.8
G-RL	94	0.968	6.4

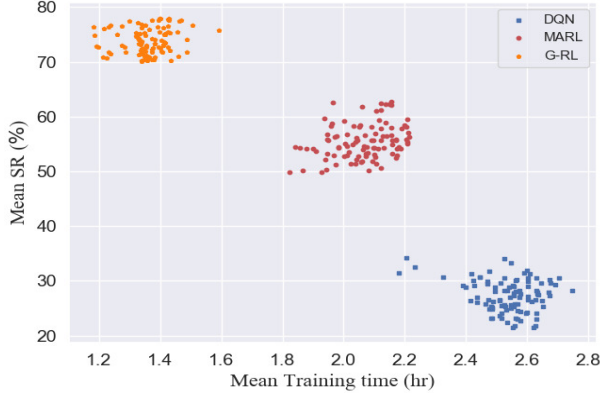


Fig. 11. Mean SR v.s. average training time for every 50 episodes in the IEEE 8500-node test system

TABLE V
TESTING RESULTS IN THE IEEE 8500-NODE TEST SYSTEM

	DQN	MARL	G-RL
Average restored load	73.53%	92.32%	100%
Average time (s)	2.35	2.05	2.02

To further demonstrate the performance, we compare the pre-trained G-RL with other RL algorithms for 50 test cases. As it may take hours to complete one case, for the sake of efficiency, we only train these algorithms for 500 episodes for each case. Fig. 11 illustrates the mean SR v.s. the average training time. The results verify that our solution is more effective than other RL algorithms.

We implement the learned control policies using different RL-based approaches, and the results are given in Table V. As expected, our solution has outstanding performance among the RL-based algorithms, benefiting from the pre-training paradigm and embedded graphical network. Hence, the results verify the ability of our solution to handle large-scale test systems.

Remark 4.1: Training GCN sometimes can be very expensive, and solving simple problems directly with model-based programs can be more straightforward and has a similar computation time as in Table IV. Instead of proposing a new method that replaces the existing solutions, we focus more on providing an alternative method when the existing solutions have difficulty in solving the problem directly with traditional programs. In particular, their performance could be degraded by three aspects: 1) the system operator has incomplete and inaccurate information; 2) the computational time for these algorithms increases significantly with the increase of controllable devices and the operation horizon and 3) these algorithms have to resolve the problem at each time step when the system condition changes with time, limiting their

flexibility. Although MILP solvers would have a comparable computational time, our RL-based method is more robust to the model inaccuracy and brings an additional advantage to solving restoration problems. Besides, with the multi-agent framework and attention mechanism, during implementation, agents only require the (latent) features from their neighbors (e.g., via communication) regardless of the number of agents, which makes our solution easily scale.

V. CONCLUSION

This paper proposes a G-RL framework to address the DSR problem. The power network is modeled by a graphical model and linked with the structure of GCNs to capture the influence of network reconfiguration and graphical interactions over controllable devices. The DRL algorithm is then adopted to learn an effective control policy for DSR using latent features produced by GCNs. The MA environment and pre-training paradigm make our solution easy to be scaled to large power systems while maintaining its efficiency. Comprehensive case studies on IEEE 123- and 8500-node test systems show that the proposed G-RL framework significantly improves the efficiency and scalability of conventional RL algorithms, e.g., DQN and MARL.

VI. ACKNOWLEDGMENT

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number 34230.

APPENDIX

A. Power flow constraints

Agents calculate the following power flow constraints to obtain R_p :

$$\sum_{h:(h,i) \in \mathcal{B}} P_{hi}^{\text{BR}} + P_i^G = \sum_{j:(i,j) \in \mathcal{B}} P_{ij}^{\text{BR}} + P_i^L, \quad (14a)$$

$$\sum_{h:(h,i) \in \mathcal{Q}} Q_{hi}^{\text{BR}} + Q_i^G = \sum_{j:(i,j) \in \mathcal{B}} Q_{ij}^{\text{BR}} + Q_i^L, \quad (14b)$$

$$U_i - U_j = 2(r_{ij} P_{ij}^{\text{BR}} + x_{ij} Q_{ij}^{\text{BR}}), \quad (14c)$$

$$-P_{ij}^{\text{max}} \leq P_{ij}^{\text{BR}} \leq P_{ij}^{\text{max}}, \quad (14d)$$

$$-Q_{ij}^{\text{max}} \leq Q_{ij}^{\text{BR}} \leq Q_{ij}^{\text{max}}, \quad (14e)$$

$$U^{\text{min}} \leq U_i \leq U^{\text{max}} \quad (14f)$$

where (14a) - (14c) are the linear DistFlow constraints [36], and (14d) - (14f) are the associated operational constraints.

B. MILP Model [13]

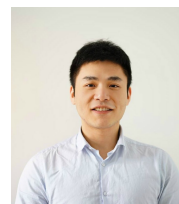
Here, we list the constraints used in the comparative study. Note that since these constraints of the fixed-time step model are directly adopted from [13], their equations are not detailed for simplicity, and we provide the corresponding equation numbers in [13]. The optimization objective is to minimize the unserved energy, subject to the following constraints:

- System model constraints (23) – (32),
- System operational constraints (37) – (39),

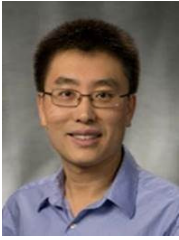
- Component operational constraints (40) – (42),
- Connectivity and sequencing constraints (44) – (50)
- Operating time constraints (51) – (52).

REFERENCES

- [1] R. Campbell, L. of Congress. Congressional Research Service, and S. Lowry, *Weather-related Power Outages and Electric System Resiliency*, ser. CRS report for Congress. Congressional Research Service, Library of Congress, 2012. [Online]. Available: <https://books.google.com/books?id=Pom3lgEACAAJ>
- [2] A. Zidan, M. Khairalla, A. M. Abdrabou, T. Khalifa, K. Shaban, A. Abdrabou, R. El Shatshat, and A. M. Gaouda, "Fault detection, isolation, and service restoration in distribution systems: State-of-the-art and future trends," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2170–2185, 2016.
- [3] J. Li, X.-Y. Ma, C.-C. Liu, and K. P. Schneider, "Distribution system restoration with microgrids using spanning tree search," *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 3021–3029, 2014.
- [4] 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, 2015.
- [5] A. Morelato and A. Monticelli, "Heuristic search approach to distribution system restoration," *IEEE Power Engineering Review*, vol. 9, no. 10, pp. 65–66, 1989.
- [6] Y.-Y. Fu and H.-D. Chiang, "Toward optimal multiperiod network reconfiguration for increasing the hosting capacity of distribution networks," *IEEE Transactions on Power Delivery*, vol. 33, no. 5, pp. 2294–2304, 2018.
- [7] C.-S. Chen, C.-H. Lin, and H.-Y. Tsai, "A rule-based expert system with colored petri net models for distribution system service restoration," *IEEE Trans. Power Syst.*, vol. 17, no. 4, pp. 1073–1080, 2002.
- [8] B. Chen, C. Chen, J. Wang, and K. L. Butler-Purry, "Sequential service restoration for unbalanced distribution systems and microgrids," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1507–1520, 2017.
- [9] S. Khushalani, J. M. Solanki, and N. N. Schulz, "Optimized restoration of unbalanced distribution systems," *IEEE Trans. Power Syst.*, vol. 22, no. 2, pp. 624–630, May 2007.
- [10] 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, 2015.
- [11] M. R. Dorostkar-Ghamsari, M. Fotuhi-Firuzabad, M. Lehtonen, and A. Safdarian, "Value of distribution network reconfiguration in presence of renewable energy resources," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 1879–1888, 2015.
- [12] E. Kianmehr, S. Nikkhal, V. Vahidinasab, D. Giaouris, and P. C. Taylor, "A resilience-based architecture for joint distributed energy resources allocation and hourly network reconfiguration," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5444–5455, 2019.
- [13] B. Chen, Z. Ye, C. Chen, and J. Wang, "Toward a milp modeling framework for distribution system restoration," *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 1749–1760, 2018.
- [14] J. Zhao, H. Wang, Y. Liu, Q. Wu, Z. Wang, and Y. Liu, "Coordinated restoration of transmission and distribution system using decentralized scheme," *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 3428–3442, 2019.
- [15] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.
- [16] E. Foruzan, L.-K. Soh, and S. Asgarpoor, "Reinforcement learning approach for optimal distributed energy management in a microgrid," *IEEE Transactions on Power Systems*, vol. 33, no. 5, pp. 5749–5758, 2018.
- [17] H. Xu, H. Sun, D. Nikovski, S. Kitamura, K. Mori, and H. Hashimoto, "Deep reinforcement learning for joint bidding and pricing of load serving entity," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6366–6375, 2019.
- [18] S. Yao, J. Gu, P. Wang, T. Zhao, H. Zhang, and X. Liu, "Resilient load restoration in microgrids considering mobile energy storage fleets: A deep reinforcement learning approach," *arXiv preprint arXiv:1911.02206*, 2019.
- [19] Y. Gao, W. Wang, J. Shi, and N. Yu, "Batch-constrained reinforcement learning for dynamic distribution network reconfiguration," *IEEE Transactions on Smart Grid*, 2020.
- [20] Q. Zhang, K. Dehghanpour, Z. Wang, F. Qiu, and D. Zhao, "Multi-agent safe policy learning for power management of networked microgrids," *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 1048–1062, 2021.
- [21] J. L. Cremer, I. Konstantelos, and G. Strbac, "From optimization-based machine learning to interpretable security rules for operation," *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 3826–3836, 2019.
- [22] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [23] W. Kool, H. van Hoof, and M. Welling, "Attention, learn to solve routing problems!" in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=ByxBFRqYm>
- [24] V. Zambaldi, D. Raposo, A. Santoro, V. Bapst, Y. Li, I. Babuschkin, K. Tuyls, D. Reichert, T. Lillicrap, E. Lockhart *et al.*, "Relational deep reinforcement learning," *arXiv preprint arXiv:1806.01830*, 2018.
- [25] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," *arXiv preprint arXiv:1802.05438*, 2018.
- [26] S. Toune, H. Fudo, T. Genji, Y. Fukuyama, and Y. Nakanishi, "Comparative study of modern heuristic algorithms to service restoration in distribution systems," *IEEE Transactions on Power Delivery*, vol. 17, no. 1, pp. 173–181, 2002.
- [27] B. Chen, Z. Ye, C. Chen, J. Wang, T. Ding, and Z. Bie, "Toward a synthetic model for distribution system restoration and crew dispatch," *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 2228–2239, 2019.
- [28] S. Sukhbaatar, R. Fergus *et al.*, "Learning multiagent communication with backpropagation," in *Advances in neural information processing systems*, 2016, pp. 2244–2252.
- [29] Y. Xu, C. Liu, Z. Wang, K. Mo, K. P. Schneider, F. K. Tuffner, and D. T. Ton, "Dgs for service restoration to critical loads in a secondary network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 435–447, 2019.
- [30] S. Poudel and A. Dubey, "Critical load restoration using distributed energy resources for resilient power distribution system," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 52–63, 2019.
- [31] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.
- [32] A. Agarwal, S. Kumar, and K. Sycara, "Learning transferable cooperative behavior in multi-agent teams," *arXiv preprint arXiv:1906.01202*, 2019.
- [33] Y. Wang, Y. Liu, W. Chen, Z.-M. Ma, and T.-Y. Liu, "Target transfer q-learning and its convergence analysis," *Neurocomputing*, 2020.
- [34] M. Riedmiller, "Neural fitted q iteration—first experiences with a data efficient neural reinforcement learning method," in *European Conference on Machine Learning*. Springer, 2005, pp. 317–328.
- [35] L. Thurner, A. Scheidler, F. Schäfer, J. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "pandapower — an open-source python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, Nov 2018.
- [36] B. Chen, C. Chen, J. Wang, and K. L. Butler-Purry, "Multi-time step service restoration for advanced distribution systems and microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6793–6805, 2018.



Tianqiao Zhao received his B.Eng. degree in automatic control from North China Electric Power University, Hebei, China, in 2013, and his PhD degree in electrical and electronic engineering from the University of Manchester, U.K., in 2019. From September 2018 to August 2019, he was a Postdoctoral Associate at Department of Electrical & Electronic Engineering, University of Manchester, UK. From September 2020 to February 2021, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering at Southern Methodist University. He is currently with Brookhaven National Laboratory as Research Associate. His research interests include distributed optimization and control, microgrid, and energy storage systems.



Jianhui Wang Dr. Jianhui Wang is a Professor with the Department of Electrical and Computer Engineering at Southern Methodist University. Dr. Wang has authored and/or co-authored more than 300 journal and conference publications, which have been cited for more than 30,000 times by his peers with an H-index of 87. He has been invited to give tutorials and keynote speeches at major conferences including IEEE ISGT, IEEE SmartGridComm, IEEE SEGE, IEEE HPSC and IGEC-XI.

Dr. Wang is the past Editor-in-Chief of the IEEE

Transactions on Smart Grid and an IEEE PES Distinguished Lecturer. He is also a guest editor of a Proceedings of the IEEE special issue on power grid resilience. He is the recipient of the IEEE PES Power System Operation Committee Prize Paper Award in 2015, the 2018 Premium Award for Best Paper in IET Cyber-Physical Systems: Theory & Applications, the Best Paper Award in IEEE Transactions on Power Systems in 2020, and the IEEE PES Power System Operation, Planning, and Economics Committee Prize Paper Award in 2021. Dr. Wang is a Clarivate Analytics highly cited researcher for production of multiple highly cited papers that rank in the top 1% by citations for field and year in Web of Science (2018-2020). He is a Fellow of IEEE.