Reconfiguring Unbalanced Distribution Networks using Reinforcement Learning over Graphs

Roshni Anna Jacob*, *Student Member*, *IEEE*, Steve Paul†, *Student Member*, *IEEE*, Wenyuan Li†, Souma Chowdhury†, *Member*, *IEEE*, Yulia R. Gel*, and Jie Zhang*, *Senior Member*, *IEEE*
*The University of Texas at Dallas, Richardson, TX 75080, USA
†University at Buffalo, Buffalo, NY 14260, USA
Email: jiezhang@utdallas.edu

***Abstract*—The recent trend in distribution system intelligence necessitates the deployment of real-time, automated, and adaptable decision-making tools. Reconfiguring the distribution network by changing the status of switches can aid in loss minimization during normal operations and resilience enhancement during disruptive events. Traditional methods employed for solving the network reconfiguration problem are model-based and scenario-specific. Besides this, the scalability and computational efficiency also limit the utilization of such techniques for online control, which could be potentially addressed by neural network based models trained with reinforcement learning (RL). To this end, we formulate the reconfiguration problem as a Markov Decision Process where the optimal control policy is learned using the RL approach. Considering the relevance of topology in decision making and the interaction between the generation and demand at different buses, we model the power distribution network along with its state variables as a graph in the learning space. Consequently, we propose an RL over graphs where a Capsule-based graph neural network is used as the policy network. The developed model is validated on the modified IEEE 13 and 34 bus test networks.**

***Index Terms*—Distribution network reconfiguration, graph neural network, reinforcement learning, topology.**

I. Introduction

The recent transformation of the power system by the smart grid concept warrants the utilization of real-time, automated, and adaptable technologies for both normal operation and emergency response. Distribution system intelligence is a facet of this change that calls for improving the efficiency and reliability of power delivery to consumers using automatic control and energy resources [1]. Another aspect of grid modernization is the rapid deployment of remote-controlled switches in the distribution network. In this context, the distribution network reconfiguration as an optimal control strategy for improving the network performance gains more relevance.

The distribution network reconfiguration (DNR) is the process of altering the topology of the distribution network by changing the status of the switches to improve the network performance [2]. There exist several objectives for DNR such as loss minimization, load balancing, minimizing voltage deviations, reliability improvement, increasing renewable penetration, and service restoration [2]–[4]. Typically, the purpose of DNR during normal operations is loss reduction for efficient and economic operations [2].

The distribution network is unbalanced with a radial operating structure. Besides this, the non-linearity in power flow, the network size, and switching decisions add to the complexity of the problem. The DNR is a non-linear, combinatorial optimization problem [5]. A plethora of work addresses the DNR problem dating back to initial heuristic approaches such as branch and bound, branch exchange, and loop elimination [5]. Over the years, two other classes for solving the optimal DNR problem have also evolved. One group consists of metaheuristic methods such as genetic algorithm [6], binary particle swarm optimization (BPSO) [7], etc.; while the other class utilizes mixed-integer programming (MIP) techniques to solve the DNR problem. Different variants of MIP formulation for DNR exist in the literature such as mixed-integer linear programming (MILP) [8], mixed-integer conic programming [9], etc.

The traditional optimal switching methods for altering the network topology rely on model-based algorithms. Considering the uncertainty in network parameters, such methods are not adaptable for online decision-making. Additionally, these methods are also limited by their scalability and computational efficiency. Over the last few years, reinforcement learning (RL) has been used for solving a wide variety of combinatorial optimization (CO), including network reconfiguration problems [10]–[12], for optimal topology control. One of the main advantage of a learning-based approach is the ability of the learned policy to generalize to unseen scenarios. Even though, a learning-based approach might not be able to generate highly optimal solutions as MILP, a learning-based approach is capable of finding an action/solution which is less optimal (as compared to MILP), but satisfying all the necessary constraints, with minimum computation time.

The power distribution network is inherently a graph with nodes (substation or load buses) and edges (lines or transformers). Hence, the voltage, current measurements, and the generation or load can be considered as data superimposed on a graph. Also, there exists a complex topological interdependency among the system variables. Consequently, the DNR can be translated as a graph learning problem. In this paper, we develop an RL framework defined in the graph domain to learn the optimal control policy of the DNR, which is modeled as a Markov Decision Process (MDP). A Capsule-based graph neural network is adopted as the policy network, for determining the switching actions depending on the state of the network. The proposed RL-based

978-1-6654-7902-8/22/$31.00

modeling approach is validated with the objective of network loss minimization, using different baseline methods such as mixed-integer second-order conic programming (MISOCP) and BPSO.

The remainder of the paper is organized as follows. In section II, the formulation of the reconfiguration problem as an MDP and in section III, the learning framework used for the problem are presented. Section IV gives a brief overview of the baseline methods used in the study. The results and discussion are presented in Section V. Finally, the conclusion and future work are discussed in Section VI.

II. Problem Formulation

The distribution feeder has prelocated switches, which can be used for opening or closing lines to improve network parameters. These switches are of two kinds: sectionalizing switches/lines that are normally closed and tie switches/lines that are normally open. The constraints associated with steady-state power flow are included in the DNR framework such as the power balance, bus voltage, and branch flow constraints. Considering the topological nature of the problem, constraints to ensure radiality in the network operating structure and the prohibition of network disintegration during normal operations are also necessary.

A. Network Reconfiguration as an MDP over Graphs

The power distribution network can be represented as a graph $\mathcal{G} = (V, E)$, where E is the set of edges connecting the set of nodes V. In the power distribution network, the buses (substation and loads) can be represented using nodes, and the lines or transformers installed between these buses can be considered as edges. The edges also include the sectionalizing and tie lines with switches. The generation and demand are the known node variables and the results of the power flow algorithm, i.e., the bus voltages and branch flows are assigned to corresponding nodes and edges on the graph. The network loss is dependent on the topology which can be altered by changing the status of the switches. This is, therefore, a binary edge classification problem in the graph domain.

The problem of reconfiguring the distribution network for loss reduction can be defined as an MDP expressed as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}_a, \mathcal{R})$. The state $\mathcal{S}$, action $\mathcal{A}$, transition probability $\mathcal{P}_a$, and reward $\mathcal{R}$ are described as follows:

- **State:** The state of the distribution network describes the operating condition of the network including the topology and system variables. This can be represented as $\mathcal{S} = [\mathcal{T}, P_D, Q_D, P_S, Q_S, P_L, V_n, I_m]$. Here, the topology of the network $\mathcal{T}$ contains the information about the existing connection between the nodes and hence the status of the edges. The active and reactive power demand at all the buses, P_D, Q_D, and the power generated by the substation (active and reactive), P_S, Q_S, also represent the system state. Other state defining variables are obtained from a distribution system power flow simulator. This includes the bus voltages V_n which are the node variables, and the branch current flow I_m which are the edge variables at the n nodes and m edges, respectively. The total network loss P_L computed from the power flow is also included in the state vector.
- **Action:** The action performed in this task is the switching of sectionalizing and tie lines. The action vector for N switches can be expressed as $\mathcal{A} = [\delta_{sw_1}, \delta_{sw_2}, ..., \delta_{sw_N}]$. The status of the switch, δ, is a binary variable with 0/1 representing open/close.
- **Transition:** The transition refers to the change of state $\mathcal{S}_{t-1}$ at time $t-1$ to a new state $\mathcal{S}_t$ at time t. The transition probability $\mathcal{P}_a$ can be expressed as $\mathcal{P}_a = Pr(\mathcal{S}_t = s' | \mathcal{S}_{t-1} = s)$.
- **Reward Function:** The reward assigned to the control action will guide the RL algorithm towards minimizing the loss in the network using switching. However, it is necessary to ensure that the control action does not perturb normal operations by disconnecting loads from the substation or changing the radial structure of the network. The reward function is hence formulated as:

$$r(s, a) = -P_L - C_p + R_p \tag{1}$$

$$C_p = \lambda_1 * (m - n + 1) + \lambda_2 * (1 - f_D) \tag{2}$$

$$R_p = \begin{cases} 0.8 & \text{if } C_p = 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In (1), the first term represents the total network loss, and the second term accounts for a penalty when the topological constraints are violated. Considering that in a radial network, the number of edges m is one fewer than the number of nodes n, the edges that result in loops are included as a penalty in (2). The term R_p is a positive reward that encourages the learning to explore more feasible region. Additionally, a binary variable f_D denotes if the network is a fully connected graph.

III. Learning Framework

In order to learn the optimal policies, we implement an on-policy RL framework with a Graph Capsule-based policy network (a.k.a GCAPS-RL) which is based on Graph Capsule Convolutional Neural Networks (GCAPCN) [13]. The Capsule-based policy network is hypothesized to incorporate local and global structural information with permutation invariance, which will then be used to compute the switching positions as a probability distribution.

A. Graph Capsule-based Policy Network

The main purpose of the encoder is to represent useful information related to a node/task as a learnable continuous vector or tensor, which will be then used to compute encodings for the switches in $\mathcal{A}$ for computing log probabilities across the action space $\mathcal{A}$. In this work, we are exploring how a GCAPCN can be implemented for learning local and global structures with the node properties, with permutation invariant node embedding. GCAPCN is a class of Graph Neural Networks (GNN), introduced in [13] to address some of the drawbacks of Graph Convolutional Neural Networks (GCN), and to enable the encoding of global information, based on **capsule networks** presented in [14]. The advantage of GCAPCN lies in capturing more local and global information,

compared to conventional aggregation operations used in GNN such as summation or standard convolution operations. The properties of node $i \in V$ is given by, $x_i = [V_a^i, V_b^i, V_c^i]$, where V is the set of all the nodes, as defined section II-A; V_a^i, V_b^i, and V_c^i are the three phase voltage for node/bus i.

Let $\mathcal{X} \in \mathbb{R}^{N_n \times |x_i|}$ be the node feature matrix, where $|x_i|$ is the input dimension for each node i. The standard graph Laplacian is defined as $\mathcal{L} = \mathcal{D} - A \in \mathbb{R}^{N_n \times N_n}$, where $\mathcal{D}$ is the degree matrix and A is the adjacency matrix of the graph. A capsule vector is computed using a Graph Capsule function based on different order of statistical moments, as shown in the equations later in this section.

We first compute a feature vector F_{0i} for each node by linear transformation of the node properties x_i, as $F_{0i} = x_i.W_0$ for all $i \in [1, N_n]$, where $W_0 \in \mathbb{R}^{|x_i| \times h_0}$, and h_0 is the length of the feature vector. For ease of representation, we are omitting the bias terms associated with all the linear transformation.

Each feature vector $F_{0i}, i \in [1, N_n]$ is then passed through a series of Graph capsule layers, where the output from the previous layers is used to compute a matrix $f_p^{(l)}(\mathcal{X}, \mathcal{L})$ using a graph convolutional filter of polynomial form as given by:

$$f_p^{(l)}(\mathcal{X}, \mathcal{L}) = \sigma(\sum_{k=0}^{K} \mathcal{L}^k (F_{(l-1)}(\mathcal{X}, \mathcal{L})^{\circ p}) W_{pk}^{(l)}) \quad (4)$$

Here $\mathcal{L}$ is the graph Laplacian, p is the order of the statistical moment, K is the degree of the convolutional filter, $F_{(l-1)}(\mathcal{X}, \mathcal{L})$ is the output from layer $l-1$, and $F_{(l-1)}(\mathcal{X}, \mathcal{L})^{\circ p}$ represents p times element-wise multiplication of $F_{(l-1)}(\mathcal{X}, \mathcal{L})$. Here, $F_{(l-1)}(\mathcal{X}, \mathcal{L}) \in \mathbb{R}^{N_n \times h_{l-1} p}$, $W_{pk}^{(l)} \in \mathbb{R}^{h_{l-1} p \times h_l}$. The variable $f_p^{(l)}(\mathcal{X}, \mathcal{L}) \in \mathbb{R}^{N_n \times h_l}$ is a matrix, where each row is an intermediate feature vector for each node $i \in [1, N_n]$, infusing nodal information from $L_e \times K$ hop neighbors, for a value of p. The output of layer l is obtained by concatenating all $f_p^{(l)}(\mathcal{X}, \mathcal{L})$, as given by:

$$F_l(\mathcal{X}, \mathcal{L}) = [f_1^{(l)}(\mathcal{X}, \mathcal{L}), f_2^{(l)}(\mathcal{X}, \mathcal{L}), \dots f_{\mathcal{P}}^{(l)}(\mathcal{X}, \mathcal{L})] \quad (5)$$

Here $\mathcal{P}$ is the highest order of statistical moment, and h_l is the node embedding length of layer l. We consider all the values of h_l (where $l \in [0, L_e]$) to be the same for this paper. Equations (4) and (5) were computed for L_e layers, where each layer uses the output from the previous layer ($F_{l-1}(\mathcal{X}, \mathcal{L})$). Adding more layers helps in learning the global structure, however, this can affect the performance by increasing the number of learnable parameters (compared to the size of the problem), leading to over-fitting. The final node embeddings are computed using a linear transformation of $F_{l=L_e}(\mathcal{X}, \mathcal{L})$.

$$F_{Nodes} = F_{l=L_e}(\mathcal{X}, \mathcal{L}).W_F \quad (6)$$

where W_F is a learnable weight matrix of size $h_{L_e}\mathcal{P} \times h_{L_e}$. For switch $\delta_i (\delta_i \in \mathcal{A})$, let (v_{i1}, v_{i2}) be the nodes (where $v_{i1}, v_{i2} \in V$) to which δ_i is connected. F_{Nodes} (of size $N_n \times h_{L_e}$) consists of the embedding s of all the N nodes, with each node embedding of length h_{L_e}. The encoding of a switch $i \in \mathcal{A}$ is computed by Eq. 7.

$$F_{switch}^i = \text{Activation}((\text{Concat}(F_N^{v_{i1}}, F_N^{v_{i2}})).W_{switch}) \quad (7)$$

where $F_{Nodes}^{v_{i1}}, F_{Nodes}^{v_{i2}} \in F_{Nodes}$, and W_{switch} is a learnable weight matrix of size $2h_{L_e} \times h_{L_e}$.

We also use the loss P_L, and the edge current I_m, before the switching action, to compute a feature vector (a.k.a context), which along with F_{switch} will be used to compute log probabilities along the action space, as explained below.

$$F_{context} = \text{Concat}(P_L, I_m).W_{context} \quad (8)$$

where $W_{context}$ is a learnable weight matrix of size $(m+1) \times h_{L_e}$.

Computing action log probabilities: The log probability of switch i being on is computed using Eq. 9.

$$LogProb_{Action}^i = (F_{switch}^i + F_{context}).W_{Action} \quad (9)$$

where W_{Action} is a learnanble weight matrix of size $h_{L_e} \times N$. Since each switch can have just two states (on or off), the probability distribution corresponding to the log probabilities computed by Eq. 9, can be considered as a Bernoulli distribution. This distribution is being used to sample the action for the Rollout operation in the RL training process.

B. Learning Algorithm

The training algorithm used here is Proximal Policy Optimization [15], with the Graph Capsule-based policy network as discussed in section III-A, and the value function being a simple feedforward network with one hidden layer and a Hyperbolic Tangent being the activation function. The training algorithm has been implemented using the Stable Baselines3 library in Python 3.7, with the settings in Table I. The embedding length h_{L_e} was taken as 128 in this paper.

TABLE I
SETTINGS FOR MODEL TRAINING

DETAILS	VALUES
Algorithm	*PPO*
Total steps	80,000
Rollout buffer size	200
Batch size	100
Optimizer	*Adam*
Learning step size	0.00001
Entropy coefficient	0.1
Value function coefficient	0.5
Epochs	100

IV. BASELINE METHODS

The DNR is modeled as a mixed-integer second-order conic programming (MISOCP) problem. Besides the variables representing the switching decisions, the variables associated with the power flow are also included in the optimization framework.

Consider the set of loads Ω_D connected at different buses in the network, the total active and reactive power demand is constrained by the power supplied by the substation P_S, Q_S as follows:

$$\sum_{i \in \Omega_D} P_i^D \leq P_S \ , \ \sum_{i \in \Omega_D} Q_i^D \leq Q_S \quad (10)$$

The power supplied by the substation is limited by the maximum rated capacity of the substation and is denoted as:

$$P_S \leq \overline{P_S} \ , \ Q_S \leq \overline{Q_S} \tag{11}$$

In the three phase branch flow equations, while applying angle relaxations [16], the square of voltage V^2 is denoted by the term U and I^2 is represented as l. For the set of buses Ω_B excluding the substation bus Ω_S, the bus voltages are constrained within upper and lower limits as:

$$\underline{U} \leq U_{i,j} \leq \overline{U}; \quad \forall i \in \Omega_B \setminus \Omega_S, \forall j \in \phi \tag{12}$$

where ϕ represents the phases (a, b, c). The voltage square at the substation (slack) bus is equated to 1.04 pu.

For the set of all elements Ω_E, and the set of switchable lines Ω_L, the active power flow through branch for non-switchable and switchable lines are constrained within limits using:

$$\underline{P}^{br}_{(k,j)} \leq P^{br}_{(k,j)} \leq \overline{P}^{br}_{(k,j)}; \quad \forall k \in \Omega_E \setminus \Omega_L, \quad \forall j \in \phi \tag{13}$$

$$\underline{P}^{br}_{(k,j)} \delta^L_k \leq P^{br}_{(k,j)} \leq \overline{P}^{br}_{(k,j)} \delta^L_k; \quad \forall k \in \Omega_L, \quad \forall j \in \phi \tag{14}$$

where δ^L_k denotes the status of switchable line k. Similar to the active power flow, the reactive power flow Q^{br} and the square of current flow l through the branches are also constrained within limits.

The equations corresponding to active and reactive power balance are formulated as follows:

$$P^{br}_{(k,j)} = P^{D}_{(N_s(k),j)} + \sum_{h \in \Omega_C(k)} P^{br}_{(h,j)} + R_{(k,j)} l_{(k,j)}; \quad \forall k \in \Omega_E, \forall j \in \phi \tag{15}$$

$$Q^{br}_{(k,j)} = Q^{D}_{(N_s(k),j)} + \sum_{h \in \Omega_C(k)} Q^{br}_{(h,j)} + X_{(k,j)} l_{(k,j)}; \quad \forall k \in \Omega_E, \forall j \in \phi \tag{16}$$

In (15) and (16), $N_s(k)$ represents the succeeding bus to which k is incident, and $\Omega_C(k)$ denotes the set of elements which are children to k.

The Kirchhoff's voltage equation is modified with the angle relaxation [16] and represented as:

$$U_{(N_s(k),j)} = U_{(N_p(k),j)} - 2(\hat{R}_{(k,j)} P^{br}_{(k,j)} + \hat{X}_{(k,j)} Q^{br}_{(k,j)}) + \hat{Z}_{(k,j)} l_{(k,j)}; \forall k \in \Omega_E \setminus \Omega_L, \forall j \in \phi \tag{17}$$

Here, $N_p(k)$ is the preceding bus at which k is incident. The parameters $\hat{R}$, $\hat{X}$, and $\hat{Z}$ are the modified resistance, reactance, and impedance of element, respectively. For a line k with switch, (17) is transformed to an inequality constraint using the big M method within $-(1-\delta^L_k)M$ and $(1-\delta^L_k)M$. Using a convex relaxation [16] on the equation for l, the second-order cone inequality constraint is formulated as:

$$l_{(k,j)} * U_{(N_p(k),j)} \geq [(P^{br}_{(k,j)})^2 + (Q^{br}_{(k,j)})^2]; \forall k \in \Omega_E, \forall j \in \phi \tag{18}$$

The spanning tree constraints to ensure that the network is radial with substation as the root node are formulated as follows:

$$\begin{aligned} \beta_{1,k} + \beta_{2,k} &= \delta^L_k; \forall k \in \Omega_L \\ \beta_{1,k} + \beta_{2,k} &= 1; \forall k \in \Omega_E \setminus \Omega_L \end{aligned} \tag{19}$$

$$\begin{aligned} \sum_{i \in \Omega_B \setminus \Omega_S} \beta_{1,\Omega_I(i)} &= 1 \\ \beta_{1,\Omega_I(i)} &= 0; \forall i \in \Omega_S \end{aligned} \tag{20}$$

where the two possible edge directions for an element are represented using β_1 and β_2. The set $\Omega_I(i)$ is used to denote the elements incident on bus i in (20). The above equations ensure that only one direction of connectivity exists on an edge and that each bus has at most one parent.

The objective function is the loss minimization, which is formulated as follows:

$$min. \sum_{k \in \Omega_E, j \in \phi} R_{(k,j)} * l_{(k,j)} \tag{21}$$

A second baseline model considered in the study is the binary particle swarm optimization (BPSO) which uses a meta-heuristic approach of optimization. An approach similar to [7] is used in this study, where a binary vector representing the optimal switch status is searched by the BPSO algorithm.

V. Results and Discussion

The model developed is validated on two modified IEEE distribution test networks, namely the 13-bus and 34-bus systems. The circuit description of these test networks built in the OpenDSS is used for power flow simulation. The OpenDSS circuit object also encapsulates the system state for each loading condition. The overall framework for implementing the topology control by switching using the GCAPS-RL is illustrated in Fig. 1. The different models are compared on an Intel Core i7-8565U 1.80GHz with 16 GB memory.

In order to understand the impact of local and global structural information for encoding, we compared GCAPS-RL with another learning-based framework, where the learning algorithm is Proximal Policy Optimization, while the policy network is a simple Multi-Layer Perceptron (MLP) network. This learning framework (a.k.a MLP-RL) has been trained with the same settings (Table I) as that for GCAPS-RL, for a fair comparison. Figures 3 and 5 show the reward per episode during training for the 13-bus and 34-bus systems, respectively, for both the methods GCAPS-RL and MLP-RL. A positive value for the reward means that the policy network is able to find switching actions without any constraint violation. The advantage of incorporating GCAPCN in the policy network is more evident for the 34-bus system (Fig. 5), where the feasible search space is smaller compared to the overall search space. Even though the MLP-RL reward is improving over time, the GCAPS-RL is able to find more solutions in the feasible search space. The training time for GCAPS-RL was found to be 70 minutes and 350 minutes for the 13-bus and 34-bus system, respectively; while for MLP-RL, the training time was found

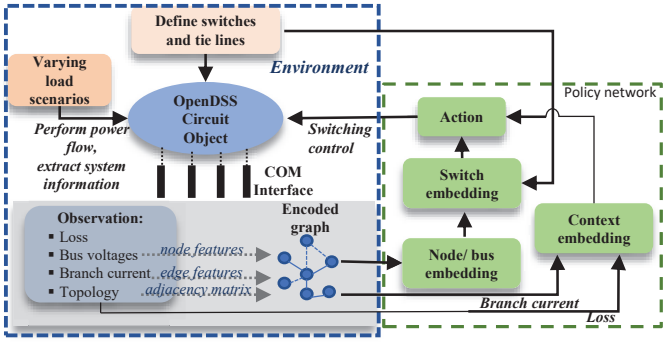

Fig. 1: Framework for implementing switching control using the GCAPS-RL

to be 60 minutes and 310 minutes for the 13-bus and 34-bus systems, respectively.

A. 13-bus Network

The IEEE 13-bus distribution test network with a rating of 3.5 MW is modified by switch placement as shown in Fig. 2. As seen in the figure, there are two sectionalizing and two tie switches. In the nominal configuration, the sectionalizing switches are closed and the tie switches are open. The resulting configuration can be represented by a vector of switch statuses as $[1, 1, 0, 0]$. This indicates that the switches 'sw1', 'sw2' are closed and 'sw3', 'sw4' are opened. The testing of the developed model and the baseline models are performed for three particular loading conditions. In the loading condition 1, the demand of load is at the rated or peak value. The loading conditions 2 and 3 are obtained by varying the demand at the loads by using a multiplication factor of 0.5 and 1.5, respectively. The optimal network configuration and the corresponding total active power loss in the network for different loading conditions are presented in the Table II. As observed in the table, considering the size of the network and the number of switches, the optimal configurations for different loading conditions do not differ. The optimal loss obtained by the baseline models and the learning-based models are compared in the table. For the MLP-RL model, however, a non entry in the loss column indicates that the corresponding configuration results in either the disintegration of the network or a non radial structure. The GCAPS-RL, on the other hand, shows no such violation of topological constraints while taking switching decisions. Besides this, as evident from the results, the GCAPS-RL also exhibits real-time decision making capability as opposed to the baseline models. The computational efficiency of the MLP-RL model is, however, negated by its inability to account for the requirements on network topology.

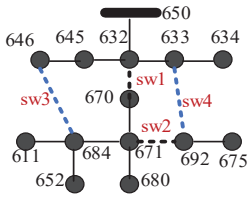

Fig. 2: Modified IEEE 13-bus distribution test network with switches. The dashed lines represent switches. The tie lines are marked in blue.

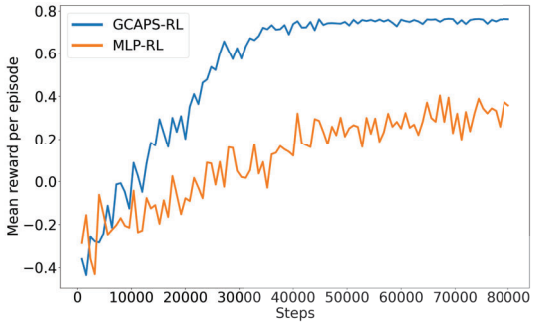

Fig. 3: **13-bus system:** Mean reward per episode during training, for GCAPS-RL and MLP-RL

B. 34-bus Network

The IEEE-34 bus distribution test network modified with switches is shown in Fig. 4. The network has a total connected load of 2.04 MW. As illustrated in the figure, this test network consists of five sectionalizing and four tie switches. The nominal configuration of the network involves closing the sectionalizing and opening the tie switches. Similar to the previous test network, we use a vector of switch statuses to denote the network configuration. For comparing the performance of the different models, three different loading conditions are considered. Load condition 1 is the scenario where the load demand is at the rated value. The loading condition 2 pertains to light loading where the demand is 0.5 times of the rated value. While the loading condition 3 considers heavy loading where the demand is 1.5 times of the rated value. Contrary to the 13-bus network, the optimal switching configuration obtained for the varying loading conditions are different. The optimal losses for the different switching decisions under varying loading condition are presented in Table II. As observed, the GCAPS-RL predicts close-to optimal loss configurations with increased speed. The MLP-RL, however, predicts configurations that result in non radial structure for loading conditions 1 and 2.

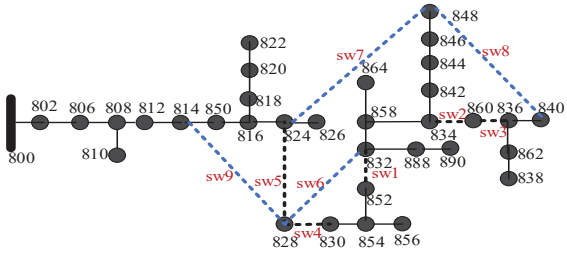

Fig. 4: Modified IEEE 34-bus distribution test network with switches. The dashed lines represent switches. The tie lines are marked in blue.

VI. Conclusion

In this paper, a reinforcement learning over graph approach has been developed for distribution network reconfiguration. The learning framework consists of an on-policy RL algorithm (Proximal Policy Optimization used here) and a policy network based on Graph Capsule Convolutional Neural Networks (GCAPCN), where GCAPCN captures the nodal properties

TABLE II
STATUS OF SWITCHES IN DISTRIBUTION NETWORKS FOR LOSS MINIMIZATION UNDER DIFFERENT LOAD CONDITIONS

Network	Method	Load condition 1		Load condition 2		Load condition 3		Mean Time (s)
		Switch status	Loss (kW)	Switch status	Loss (kW)	Switch status	Loss (kW)	
13-bus	GCAPS-RL	[0, 1, 0, 1]	157.90	[0, 1, 0, 1]	34.29	[0, 1, 0, 1]	360.28	0.0019
	MLP-RL	[0, 1, 0, 0]	-	[0, 1, 0, 1]	34.29	[1, 1, 0, 1]	-	0.0009
	BPSO	[1, 0, 0, 1]	105.73	[1, 0, 0, 1]	24.61	[1, 0, 0, 1]	251.76	25.5710
	MISOCP	[1, 0, 0, 1]	105.73	[1, 0, 0, 1]	24.61	[1, 0, 0, 1]	251.76	0.2452
34-bus	GCAPS-RL	[1, 1, 0, 1, 0 0, 1, 1, 0]	58.51	[1, 1, 0, 1, 0 0, 1, 1, 0]	24.08	[1, 1, 0, 1, 0 0, 1, 1, 0]	124.18	0.0020
	MLP-RL	[1, 1, 1, 1, 0 0, 1, 1, 0]	-	[1, 1, 1, 1, 0 0, 1, 1, 0]	-	[1, 1, 0, 1, 0 0, 1, 1, 0]	124.18	0.0010
	BPSO	[1, 1, 1, 0, 1, 1, 0, 0, 0]	56.66	[1, 0, 1, 0, 1, 1, 0, 1, 0]	22.38	[0, 1, 1, 1, 0, 1, 0, 0, 1]	110.11	245.398
	MISOCP	[1, 1, 1, 0, 1, 1, 0, 0, 0]	56.66	[0, 0, 1, 1, 0, 0, 1, 1, 1]	22.48	[0, 1, 1, 1, 0, 1, 0, 0, 1]	110.11	0.5654

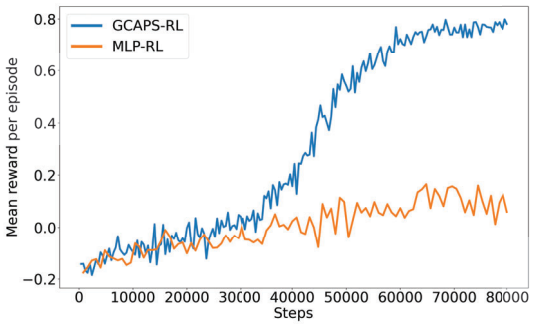

Fig. 5: **34-bus system:** Mean reward per episode during training, for GCAPS-RL and MLP-RL

as well as the local and global structural information of the nodes. For validating the proposed framework, loss minimization during normal operation under varying load conditions has been considered for two different test networks, the 13-bus and the 34-bus systems. The proposed model has demonstrated close to optimal decision-making in real time. In addition to the online switching control, we have shown that the topological inter-dependencies of the various buses and constraints on the operating structure are well captured by the graph-based learning model. The proposed model is thus an important step towards automating the topological control of the distribution network.

Along with demonstrating the proposed approach's scalability to larger grid systems, future extensions of the proposed approach could include modeling decisions for service restoration during network outages. In that case, the benefits of the model including its real-time response can be realized during emergency conditions, as well as integration of network shape descriptors delivered by tools of persistent homology.

ACKNOWLEDGMENT

This material is based upon work sponsored by the Department of the Navy, Office of Naval Research under ONR award number N00014-21-1-2530. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research.

REFERENCES

[1] "Distribution intelligence," https://www.smartgrid.gov/the_smart_grid/distribution_intelligence.html.
[2] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Power Engineering Review*, vol. 9, no. 4, pp. 101–102, 1989.
[3] R. A. Jacob and J. Zhang, "Distribution network reconfiguration to increase photovoltaic hosting capacity," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2020, pp. 1–5.
[4] R. A. Jacob and J. Zhang, "Outage management in active distribution network with distributed energy resources," in *2020 52nd North American Power Symposium (NAPS)*. IEEE, 2021, pp. 1–6.
[5] R. J. Sarfi, M. Salama, and A. Chikhani, "A survey of the state of the art in distribution system reconfiguration for system loss reduction," *Electric Power Systems Research*, vol. 31, no. 1, pp. 61–70, 1994.
[6] B. Radha, R. T. King, and H. C. Rughooputh, "A modified genetic algorithm for optimal electrical distribution network reconfiguration," in *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, vol. 2. IEEE, 2003, pp. 1472–1479.
[7] B. Amanulla, S. Chakrabarti, and S. Singh, "Reconfiguration of power distribution systems considering reliability and power loss," *IEEE transactions on power delivery*, vol. 27, no. 2, pp. 918–926, 2012.
[8] H. M. Ahmed and M. M. Salama, "Energy management of ac–dc hybrid distribution systems considering network reconfiguration," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4583–4594, 2019.
[9] 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.
[10] Y. Gao, W. Wang, J. Shi, and N. Yu, "Batch-constrained reinforcement learning for dynamic distribution network reconfiguration," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5357–5369, 2020.
[11] Y. Gao, J. Shi, W. Wang, and N. Yu, "Dynamic distribution network reconfiguration using reinforcement learning," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2019, pp. 1–7.
[12] O. B. Kundačina, P. M. Vidović, and M. R. Petković, "Solving dynamic distribution network reconfiguration using deep reinforcement learning," *Electrical Engineering*, 2021.
[13] S. Verma and Z. L. Zhang, "Graph capsule convolutional neural networks," 2018.
[14] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Artificial Neural Networks and Machine Learning – ICANN 2011*, T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 44–51.
[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.
[16] M. Farivar and S. H. Low, "Branch flow model: Relaxations and convexification—part i," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2554–2564, 2013.