Distribution Grid Critical Load Restoration under Uncertain Topology Changes via a Hierarchical Multi-Agent Reinforcement Learning Approach

Vanshaj Khattar, Yiyun Yao, *Member, IEEE*, Fei Ding, *Senior Member, IEEE*, Ming Jin, *Member, IEEE*

***Abstract*—Extreme weather events and/or cyber-attacks can significantly disrupt the power generation of a power grid, leading to catastrophic consequences. In this paper, the critical load restoration (CLR) problem in the community distribution grid is addressed. Existing approaches for CLR rely on the assumption that the grid topology does not change during the restoration period. However, it is highly likely that, under major disruptions, some of the buses/lines can get disconnected and change the underlying topology of the grid. These uncertain topology changes could lead to a different CLR problem formulation as well as a restoration strategy. To this end, we propose a hierarchical multi-agent reinforcement learning (HMARL) framework for CLR, which uses *topology-dependent action masks* (TDAM) to handle the changing topology. The main idea is to divide the distribution grid into multiple cells capable of independent control and a coordinating agent to allow power transfer between different cells under topological variations during restoration. Moreover, TDAM helps identify the actions that are unavailable after the topology change. We demonstrate the effectiveness of the proposed method on a modified IEEE-123 bus system, showing that it achieves robust load restoration despite fluctuating topology.**

***Index Terms*—Critical load restoration, distribution grid, multi-agent reinforcement learning, topology uncertainties.**

I. Introduction

The increasing power demands due to rapid urbanization have led to increased stress on our power grids, which can lead to power outages [1]. Moreover, extreme weather events and/or cyber-attacks make power grids more vulnerable to power outages. The increasing integration of distributed energy resources (DERs) allows power backup during outage events, improving grid resiliency. Devices such as battery energy storage systems (BESS) and solar PV panels, equipped with grid-forming and grid-following inverters, can support loads and maintain power supply during grid failures by acting as decentralized power sources [2]. Effective management and coordination of these DERs are essential to maximize their potential for load restoration under power outages.

Traditional approaches for critical load restoration (CLR) involve formulating an optimal power flow (OPF) problem to achieve coordinated control over a restoration horizon [3]. For example, in [4], the authors formulate the load restoration problem as a mixed integer non-linear program. In [5], the authors formulate the restoration problem using the alternating direction method of multipliers (ADMM) algorithm, where they model the nonlinearities from three-phase unbalanced power flow and distribution components using a convex quadratic programming model. However, the OPF problems are non-convex and NP-hard to solve, where the existing solutions rely on the convex surrogates, leading to approximation errors. Moreover, the inherent uncertainty in the renewable DERs generation increases the complexity during real-time control. Stochastic and robust optimization approaches have been proposed to overcome these limitations [6] but still suffer from high computational complexity during real-time implementation.

Recently, reinforcement learning (RL)–based solutions have emerged as an alternative to solve the load restoration problem due to their ability to adapt to the changing conditions [7], [8]. In RL-based approaches, the CLR problem is formulated as a Markov Decision Process (MDP), where the agent continuously interacts with the environment/MDP to compute the optimal set of actions (policy). The main advantage of the RL-based approach is that it allows the policy to be trained offline before the deployment, allowing for fast, real-time implementation during the outage event.

Despite the recent successes of RL-based and optimization-based approaches, none of the existing solutions can handle the cases where the underlying topology of the microgrid changes. Topology changes typically involve the loss of microgrid elements (lines or buses) due to an extreme event. It can trigger a cascade of events involving sectionalized switches and fault location, isolation, and service restoration (FLISR) mechanisms. These events can significantly alter the system's topology, potentially disconnecting entire portions of the grid, including lines, loads, and DERs. This further leads to a different action space for the RL formulation of the CLR due to disconnected DERs. Consequently, the system's generation and load restoration capabilities are also directly impacted, affecting the overall performance and rewards associated with the RL formulation for the CLR problem.

In this paper, we address the problem of CLR under topology changes using a **hierarchical multi-agent RL (HMARL)** framework. HMARL consists of two main components. The first component of HMARL is hierarchical RL structure, where, after the power outage, the distribution system is

This work was authored in part by the National Renewable Energy Laboratory for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. The work is partially supported by U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Solar Energy Technologies Office Agreement Number 37770. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

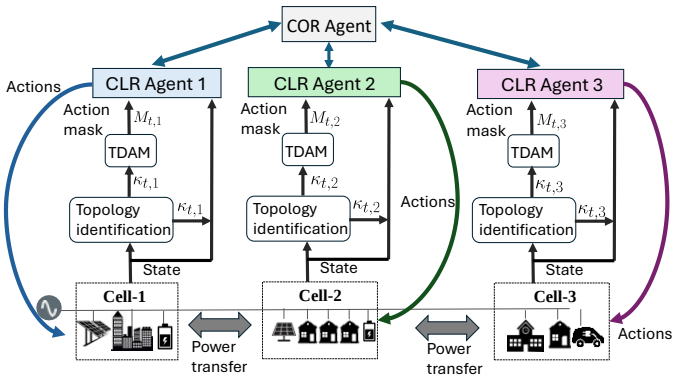


Fig. 1: The proposed hierarchical multi-agent RL method for CLR. It consists of a topology identification module that outputs the current switch status $\kappa_{t,c}$ for each cell, which is further used to construct action masks $M_{t,c}$ for each cell.

divided into multiple community microgrids, which we refer to as *cells*. During the load restoration period, each cell disconnects from the main grid and is controlled by its RL agent to restore the critical loads. While recent works have explored decentralized MARL for CLR, there is no coordination among different agents to restore the load. We propose a hierarchical RL component consisting of another coordinating agent, referred to as the COR agent, which controls the power flow between different cells. This splitting of the entire network into distinct cells with power flow coordination among them allows for a robust and shared response to topology changes during the restoration period. The second component of HMARL is the **topology dependent action masks (TDAM)** to handle the changing action spaces under changing topology, which informs the agents of the unavailable actions as the topology changes (see Section III-B). The overall learning architecture is shown in Fig. 1. The main contributions of this work are:

1) We propose a novel hierarchical multi-agent RL framework for CLR, which can handle uncertain topology changes during the restoration period.
2) We propose the topology-dependent action masking to handle the dynamic action spaces of the RL agents during topology changes.
3) Finally, we validate the effectiveness of the proposed method on a modified IEEE-123 bus system and show that the proposed method can restore critical loads even under contingencies, including topology changes.

A closely related work of [9] proposes real-time outage management, where the authors consider topology changes during test time. However, their method is developed for distributed network reconfiguration, which does not change the action space after topology changes. In contrast, our approach is concerned with load restoration using different DERs and also faces the additional challenge of changing action space.

II. Problem Formulation

A. CLR Problem Formulation

The goal of the critical load restoration problem is to restore as many critical loads as possible during the power outage duration. Let T be the total outage duration, where we use $t \in \mathcal{T} = \{1, \dots, T\}$ to denote the discrete time steps of the outage duration and $\mathcal{L}$ to denote the set of all loads. As we divide the microgrid into different cells, we denote each cell by $c \in \mathcal{C} = \{1, \dots n^c\}$, where the criticality of each load is denoted by an importance factor $z^i (i \in \mathcal{L})$. The total number of loads is denoted by $N = |\mathcal{L}| = \sum_{c \in \mathcal{C}} N_c$, where N_c is the number of loads in cell c. A load i is called critical if $z_i \geq z^{th}$ for some threshold z^{th}. The load request at time t is denoted as $\mathbf{p}_{t,c}^{\mathcal{L},\text{req}} := \mathbf{p}_{t,c}^{\text{CL,req}} + \mathbf{p}_{t,c}^{\text{NL,req}}$ which collects both critical and non-critical load request at time t in cell c. We denote the set of PV generators by $\mathcal{H}$, battery energy storage systems (BESS) as $\mathcal{E}$, and fuel-based generators as $\mathcal{D}$, and all DERs combined as $\mathcal{G} := \mathcal{H} \cup \mathcal{E} \cup \mathcal{D}$.

The control actions for each RL agent in cell c is denoted as $x_t := [\mathrm{p}_t^{\mathcal{G}}, \mathrm{q}_t^{\mathcal{G}}, \mathrm{p}_t^{\mathcal{L}}, \mathrm{q}_t^{\mathcal{L}}, \mathrm{w}_t^{\mathcal{C}}, \mathrm{p}_t^{\mathcal{C}}, \mathrm{q}_t^{\mathcal{C}}]^\top$, where $\mathrm{p}_t^{\mathcal{G}}, \mathrm{q}_t^{\mathcal{G}}$ denotes the DER power set points, $\mathrm{p}_t^{\mathcal{L}}, \mathrm{q}_t^{\mathcal{L}}$ denotes the amount of restored load, and $\mathrm{w}_t^{\mathcal{C}}, \mathrm{p}_t^{\mathcal{C}}, \mathrm{q}_t^{\mathcal{C}}$ denotes the switch status and the power transfer among different cells, respectively. The overall load restoration problem can be formulated as follows:

$$\max_{x_t} \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}} \left(r_{t,c}^{\text{CLR}} - v_{t,c} \right) \tag{1a}$$

$$\text{s.t.} \quad r_{t,c}^{\text{CLR}} = z_c^\top \mathrm{p}_{t,c}^{\mathcal{L}} - \ell_{t,c}, \quad \forall c \in \mathcal{C}, \forall t \in \mathcal{T} \tag{1b}$$

$$\ell_{t,c} = z_c^\top \text{diag}\{\epsilon_c\} \mathbf{H}_{t,c} \left[\underbrace{\mathrm{p}_{t-1,c}^{\text{CL}} - \mathrm{p}_{t,c}^{\text{CL}}}_{\text{absolute change}} \right]^+, \tag{1c}$$

$$\mathbf{H}_{t,c} = \text{diag} \left\{ H(\underbrace{\tilde{\mathrm{p}}_{t-1,c}^{\text{CL}} - \tilde{\mathrm{p}}_{t,c}^{\text{CL}}}_{\text{relative change}}) \right\}, \tag{1d}$$

$$f^{\text{pf}} \left(\mathrm{p}_{t,c}^{\mathcal{G}}, \mathrm{q}_{t,c}^{\mathcal{G}}, \mathrm{p}_{t,c}^{\mathcal{L}}, \mathrm{q}_{t,c}^{\mathcal{L}}, \mathrm{w}_t^{\mathcal{C}}, \mathrm{p}_t^{\mathcal{C}}, \mathrm{q}_t^{\mathcal{C}} \right) = 0, \tag{1e}$$

$$v_{t.c} = \lambda^v \| \left[\mathrm{v}_{t,c} - \overline{\mathrm{v}} \right]^+ + \left[\underline{\mathrm{v}} - \mathrm{v}_{t,c} \right]^+ \|_2^2, \tag{1f}$$

$$f^{\text{bal}} \left(\mathrm{p}_{t,c}^{\mathcal{G}}, \mathrm{q}_{t,c}^{\mathcal{G}}, \mathrm{p}_{t,c}^{\mathcal{L}}, \mathrm{q}_{t,c}^{\mathcal{L}}, \mathrm{w}_t^{\mathcal{C}}, \mathrm{p}_t^{\mathcal{C}}, \mathrm{q}_t^{\mathcal{C}} \right) = 0, \tag{1g}$$

$$\sum_{t \in \mathcal{T}} p_t^f \tau \leq E^f, f \in \mathcal{D}, \tag{1h}$$

$$S_{t+1}^b = f^{\text{bess}}(S_t^b, p_t^b), \underline{S^b} \leq S_t^b \leq \overline{S^b}, b \in \mathcal{E}, \tag{1i}$$

$$g(\mathrm{w}_t^{\mathcal{C}}) \geq 0, \tag{1j}$$

$$\underline{\mathrm{p}_t^e} \leq \mathrm{p}_t^e \leq \overline{\mathrm{p}_t^e}, \underline{\mathrm{q}_t^e} \leq \mathrm{q}_t^e \leq \overline{\mathrm{q}_t^e}, e \in \{\mathcal{G}, \mathcal{L}, \mathcal{C}\}, \tag{1k}$$

where $r_{t,c}^{\text{CLR}}$ and $v_{t,c}$ denote the load restoration reward and the voltage violation penalty at time t in cell c. The first term of $r_{t,c}^{CLR}$ is proportional to the restored load, $\mathrm{p}_{t,c}^{\mathcal{L}}$ to promote higher load restoration, while the second term, $\ell_{t,c}$ penalizes the previously restored critical loads. In (1d), $\tilde{\mathrm{p}}_{t,c}^{\text{CL}}$ denotes the relative load restoration level which is the element-wise ratio between $\mathrm{p}_{t,c}^{\mathcal{L}}$ and $\mathrm{p}_{t,c}^{\text{CL,req}}$. $H(\cdot)$ is the element-wise Heaviside step function and is used to select the loads whose relative load restoration level is dropped from the previous step. In (1c), Power flow and power balance constraints are shown in (1e) and (1g). The voltage violations are captured in (1f), where $\mathrm{v}_{t,c}$ denotes the bus voltages in cell c, λ^v is the penalty factor with $\overline{\mathrm{v}}$ and $\underline{\mathrm{v}}$ denoting the upper and lower bounds, respectively. All DERs have maximum available fuel E^f as shown in (1h), where τ is the control interval, and (1i) shows the BESS

state of charge S_t^b limits. Finally, (1j) and (1k) denote the radial topology and operational constraints. In this study, we focus mainly on scheduling DERs and load pick-up, and their dynamic performance is not considered for simplicity.

B. MARL Formulation for Load Restoration

We first define the observation and action spaces of CLR and the COR agents.

Observation space for the CLR agents: $\mathrm{o}_{t,c}^{\mathrm{CLR}} := [\mathrm{p}_{t,c}^{\mathcal{H},\mathcal{K}}, \mathrm{b}_{t,c}^{\mathcal{E}}, \mathrm{d}_{t,c}^{\mathcal{D}}, \mathrm{p}_{t,c}^{\mathcal{L},\mathrm{req}}, \mathrm{p}_{t-1,c}^{\mathcal{L}}, \kappa_{t,c}, t]^\top$, where $\mathrm{p}_{t,c}^{\mathcal{H}}$ is the K-step look ahead forecast of the PV power generation from time t, $\mathrm{b}_{t,c}^{\mathcal{E}}$ is the SOC of all BESS in cell c, $\mathrm{d}_{t,c}^{\mathcal{D}}$ denotes the remaining fuel in the generators, $\mathrm{p}_{t,c}^{\mathcal{L},\mathrm{req}}$ and $\mathrm{p}_{t-1,c}^{\mathcal{L}}$ denote the the current load requests and the load restored at previous time step, respectively. The current topology information for cell c at time t is denoted by $\kappa_{t,c}$ which is obtained from the topology identification module, as explained in Section III-A.

Action space for the CLR agents: $\mathrm{a}_{t,c}^{CLR} := \left[\mathrm{p}_{t,c}^{\mathcal{H}}, \mathrm{q}_{t,c}^{\mathcal{H}}, \mathrm{p}_{t,c}^{\mathcal{E}}, \mathrm{p}_{t,c}^{\mathcal{D}}, \mathrm{q}_{t,c}^{\mathcal{D}}, \mathrm{p}_{t,c}^{\mathcal{L}}\right]^\top$, where $\mathrm{p}_{t,c}^{\mathcal{H}}$ and $\mathrm{q}_{t,c}^{\mathcal{H}}$ denote the power dispatch for the PV inverter set points, $\mathrm{p}_{t,c}^{\mathcal{E}}$ is the power charge/discharge for the BESS, $\mathrm{p}_{t,c}^{\mathcal{D}}$ and $\mathrm{q}_{t,c}^{\mathcal{D}}$ are the power dispatch for the fuel-based generators, $\mathrm{p}_{t,c}^{\mathcal{L}}$ is the current load restoration decision.

Observation space for the COR agent: $\mathrm{o}_t^{COR} := \left[\mathrm{o}_{t-1,c}^{\mathrm{cell}}|_{c\in\mathcal{C}}, \mathrm{w}_{t-1}^{\mathcal{C}}\right]$, where $\mathrm{o}_{t-1,c}^{\mathrm{cell}} := \left[\mathrm{p}_{t-1,c}^{\mathcal{G}}, \mathrm{p}_{t-1,c}^{\mathrm{CL,req}}, \mathrm{p}_{t-1,c}^{\mathrm{CL}}\right]$ denotes the total DER generation, total critical load request and total critical load restored in cell c at $t-1$, and $\mathrm{w}_{t-1}^{\mathcal{C}}$ denotes the switch status at previous time step.

Action space for the COR agent: $\mathrm{a}_t^{COR} := \left[\mathrm{w}_t^{\mathcal{C}}, \mathrm{p}_t^{\mathcal{C}}\right]$, where $\mathrm{w}_t^{\mathcal{C}}$ are the switch status at time t between all cells, and $\mathrm{p}_t^{\mathcal{C}}$ are the power transfer among different cells.

Reward formulation. We define the total average reward (TAR) at time t for each agent as

$$r_t = \frac{\sum_{c\in\mathcal{C}} r_{t,c}^{\mathrm{CLR}} + v_{t,c} - \lambda^{\mathrm{bal}}|f_{t,c}^{\mathrm{bal}}|}{(n^c+1)}. \tag{2}$$

where the last term $\lambda^{\mathrm{bal}}|f_{t,c}^{\mathrm{bal}}|$ penalizes the power imbalance, where $|f_{t,c}^{\mathrm{bal}}|$ corresponds to (1g). Constraint (1e) is enforced by the OpenDSS simulator as a part of the Gym environment, constraint (1j) is enforced by the COR agent, and the constraint (1k) is enforced by the CLR action space design.

III. Methodology

A. Real-time Topology Identification

The topology information of each cell is never directly observable. The only observable states are the voltage and power measurements at each bus location. Therefore, to get the current topology state for each cell $\kappa_{t,c}$, we train a multi-layer perceptron (MLP) on the extracted features from the voltage and power measurements, as done in [10]. The main idea is to extract the features that can capture the connection relationships between any two nodes from the nodal voltage and power measurements. We use the same three features as used in [10] to train the MLP model:

1) **Voltage correlation feature:** The first feature is the Pearson correlation coefficient of the voltage time-series model between two nodes. The main idea is that the nodes which are connected together have similar voltage fluctuation profiles.
2) **Voltage-drop fluctuation feature:** The second feature is the standard deviation of the voltage-drop time series between any two connected nodes.
3) **Power flow feature:** The third feature is the coefficient of determination between the power measurements of any two adjacent nodes connected by a closed switch.

All three features are extracted from time-series data of the nodal voltages and power measurements, which are then used to construct the training, validation, and testing data. The labels in the time-series data are either 1 or 0, representing close and open switch states, respectively. Finally, the trained MLP model is used for real-time topology identification $\kappa_{t,c}$.

B. Handling Dynamic Action Space via Action Masking

One of the challenges stemming from the changing topology in the load restoration problem is to handle the dynamic action space due to disconnected DERs. Therefore, we propose a topology-dependent action masking (TDAM) technique to handle the dynamic action space for each of the CLR agents. The main idea is that, in addition to storing states, actions, rewards, and the next state in the replay buffer, we also pass the next unavailable actions mask, denoted as $M_{t,c}$ at time t in cell c. This action mask is determined from the real-time topology identification $\kappa_{t,c}$ described in Section III-A. For example, if a CLR agent has 5 DERs at time step t, and if at $t+1$, the fourth DER gets disconnected, the action mask will change from $[1,1,1,1,1]$ to $[1,1,1,0,1]$, where 1 indicates an active action, and 0 indicates the unavailable action. The action mask information is then used by each of the CLR agents to set the corresponding Q-value corresponding to the action with the zero mask as negative infinity. Therefore, when the actor maximizes the Q-function to find the optimal action, the agent does not take the action with the zero mask. The main advantage of using TDAM is that it prevents us from handling different dimensions of the action space at different time steps, which would require different policy networks for each possible topology change.

C. Hierarchical Multi-Agent Reinforcement Learning

Now, we describe the overall learning architecture of HMARL. We train both the CLR and COR agents using MARL. The objective of MARL training is to search for a joint policy parameter θ^{a*} that maximizes the reward objective:

$$\begin{aligned}\theta^{a*} &= \arg\max_{\theta^a} J(\theta^a) \\ &= \arg\max_{\theta^a} \mathbb{E}_{a_t=\Pi_{i\in\mathcal{I}}\pi_{\theta^i}(o_t^i)}\left[\sum_{t\in\mathcal{T}} \overline{\mathcal{R}}(s_t, a_t, M_{t,c})\right],\end{aligned} \tag{3}$$

where $M_{t,c}$ is the action mask at time t for cell c. We follow *centralized-training and decentralized-execution* (CTDE) for the multi-agent learning as proposed in the MADDPG paper [11]. The CTDE uses a central critic for the overall environment instead of independent critics for each environment,

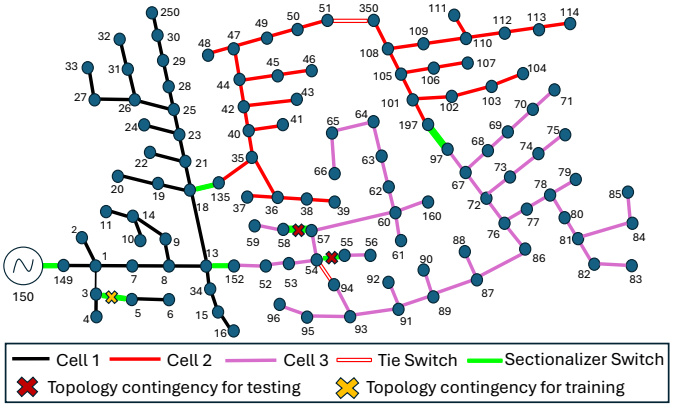


Fig. 2: The modified IEEE-123 bus system.

TABLE I: DER and load details for the IEEE-123 bus system

Cell	Total load (kW)	Critical load (kW)	PV (kW)	BESS (kW/kWh)	Fuel-based generator (kW)
1	760	340	240	220/1100	175
2	1075	240	390	180/900	280
3	1100	295	390	240/1200	210

leading to a more accurate value function estimation. Central critics have access to the observations and actions for all the agents during training, i.e., $Q_i^{\Pi_i \in \mathcal{I} \pi_{\theta^i}}\left(o^1, \ldots, o^n, a^1, \ldots, a^n\right)$. Once the agents are trained using the central critic, local control policies π_{θ^i} are plugged into agents for decentralized execution with only local observations, $o_{t,c}$.

IV. Case Study

A. System Setup

We validate the effectiveness of the proposed HMARL method on a modified IEEE-123 bus system with 2 tie-switches and 8 sectionalizer switches. We model the distributed microgrid system using the OpenDSS simulator. First, we divide the bus system into three cells, where the power transfer can happen between each cell through sectionalizer switches. The total number of loads in each cell are 23, 24, and 28, respectively, and critical loads are 13, 21, and 18, respectively. The number of BESS in each cell is 11, 8, and 10, and the number of fuel generators is 5, 7, and 6, respectively, where each DER is allotted sequentially to each bus in each cell. We use an importance factor of $z^i = 1$ to denote a critical load and an $z^i = 0.1$ to denote a non-critical load. Figure 2 shows the whole 123-bus system, with each cell denoted with a different color of buses. We consider a 3-day load restoration period, with a 15-minute control interval, making one episode a total of 288 steps. Load and PV shapes are collected from a real feeder located in western Colorado for the year 2022.

B. Training and Testing

We use MADDPG [11] for the HMARL training. Specifically, we use the XuanCe [12] for training, an open-source repository for MARL algorithm implementations. To simulate the topology changes, we only consider the opening and closing of the tie/sectionalizer switches. We train the agent using three different strategies: *1)* No topology change encountered during training; *2)* 1 topology change in cell 1 encountered at fixed time step; *3)* 1 topology change in cell 1 encountered at random time steps. We denote each strategy as HMARL1, HMARL2, and HMARL3, respectively. During training, we only consider the sectionalizer switch between buses 3 and 5 going from close to open, thus disconnecting the loads and DERs on buses 5 and 6 (yellow cross in Figure 2).

TABLE II: Restored reward during testing for scenario S1

	HMARL1	HMARL2	HMARL3
Cell 1 load restored reward	0.391	0.387	**0.397**
Cell 2 load restore reward	0.214	0.282	**0.301**
Cell 3 load restored reward	0.257	0.451	**0.544**
Total load restored reward	0.862	1.120	**1.242**

TABLE III: Restored reward during testing for scenario S2

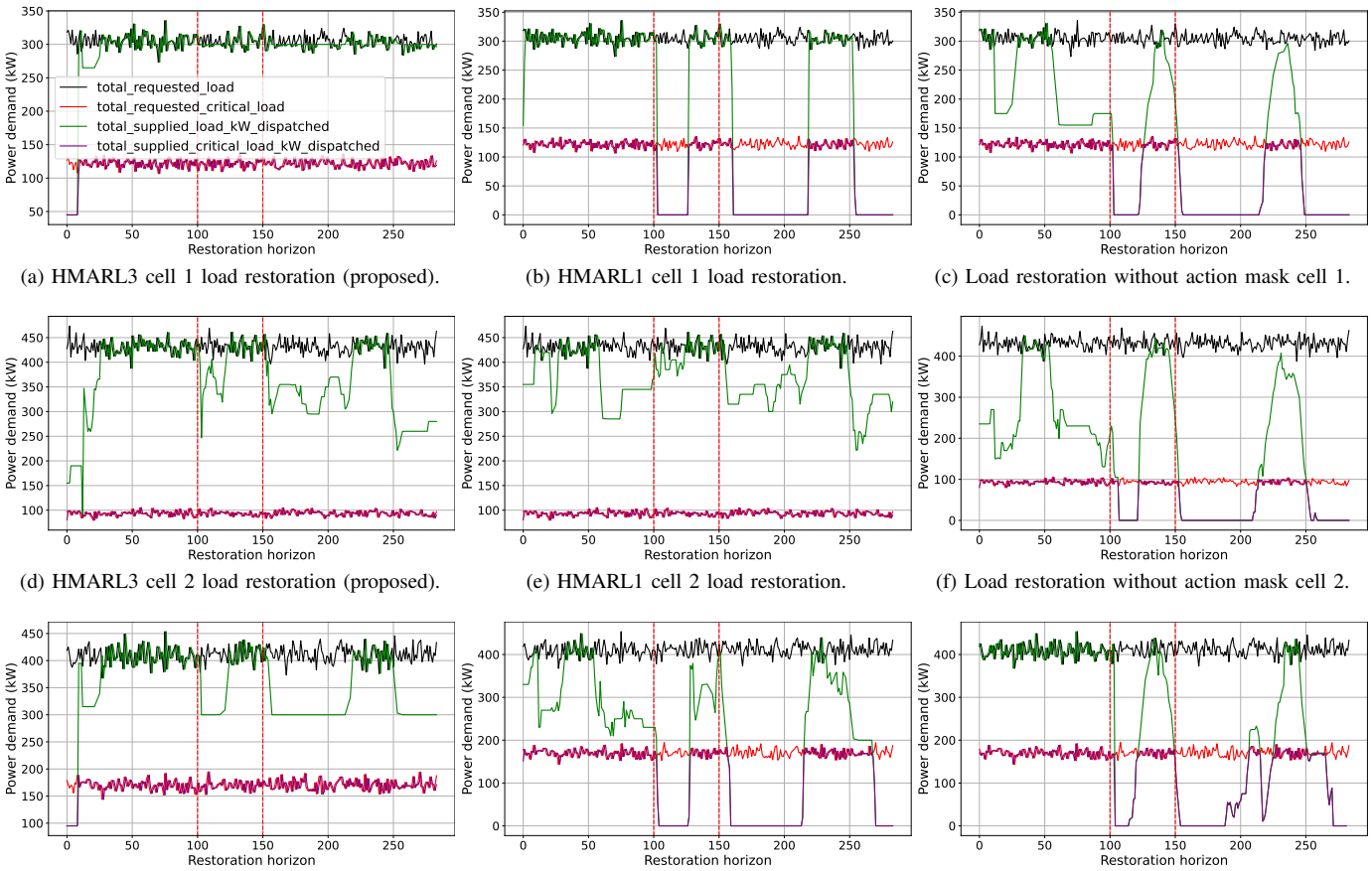
	HMARL1	HMARL2	HMARL3
Cell 1 load restored reward	0.201	0.233	**0.301**
Cell 2 load restore reward	0.360	0.350	**0.453**
Cell 3 load restored reward	0.461	0.451	**0.480**
Total load restored reward	1.022	1.034	**1.234**

We test the agent in two scenarios: S1, where topology changes occur at time steps 100 (switch 6 opens up) and 150 (switch 10 opens up), and S2, where the same changes occur at time steps 50 and 200. Both contingencies take place in cell 3 (red crosses in Fig. 2).

Topology identification results. First, we report the detection accuracy of the topology identification module trained using the procedure described in Section III-A. The proposed topology identification method is able to achieve 99.6% accuracy on the test scenarios of predicting the switch states as open or closed. Therefore, the trained MLP can be used as a reliable estimator for switch status $\kappa_{t,c}$ in each cell to determine the topology-dependent action masks $M_{t,c}$.

C. Simulation Results

The results of the restored load reward r_t^{CLR} for all three methods for both scenarios S1 and S2 are shown in Table II and III. We can observe that the HMARL3, where we expose the RL agents to random time topology changes during training, leads to a more generalizable policy that is able to adapt better to an unseen topology change at the test time for both S1 and S2. In Figure 3, we show the load restoration performance for HMARL3, HMARL1, and the HMARL without TDAM. We can see from Figures 3a, 3d, and 3g that at time steps 100 and 150 for cell 3, the total supplied load in kW starts dropping suddenly due to sudden disconnection of the DERs. However, the COR agent helps in power transfer between cell 2 and cell 3, as seen from the decrease in the restored load in cell 2 after time steps 100 and 150. Moreover, all of the CLR agents maintain the critical load demands in each of the cells. The HMARL1 agent that did not encounter any topology changes during training has worse performance and is not able to adapt to the topology changes during the test time (see Figures 3b, 3e, and 3h). We also show the impact of training the agent without TDAM. We can see from Figures 3c, 3f, and 3i, that the HMARL without TDAM is not able to adapt to the topology changes at the test time,



(a) HMARL3 cell 1 load restoration (proposed). (b) HMARL1 cell 1 load restoration. (c) Load restoration without action mask cell 1.

(d) HMARL3 cell 2 load restoration (proposed). (e) HMARL1 cell 2 load restoration. (f) Load restoration without action mask cell 2.

(g) HMARL3 cell 3 load restoration (proposed). (h) HMARL1 cell 3 load restoration. (i) Load restoration without action mask cell 3.

Fig. 3: The figures above show the load restoration performance in each of the three cells for HMARL3 (proposed), HMARL1, and without action masks for scenario S1. The red vertical lines show the topology change time steps in cell 3.

as seen from the critical load restoration performance after time steps 100 and 150. The changed action space in one cell drastically deteriorates the performance in all the cells. This highlights the advantage of incorporating TDAM and the COR agents for efficient critical load restoration.

V. Conclusion

We proposed a multi-agent hierarchical RL framework for critical load restoration. The proposed method can adapt to the topology changes in the distribution grid during the restoration period. The main ideas were to split the system into different cells and use real-time topology identification with action masking to handle the continuously changing action space.

References

[1] Jiancun Liu, Chao Qin, and Yixin Yu. A comprehensive resilience-oriented flisr method for distribution systems. *IEEE Transactions on Smart Grid*, 12(3):2136–2152, 2020.

[2] Amir H Etemadi, Edward J Davison, and Reza Iravani. A decentralized robust control strategy for multi-der microgrids—part i: Fundamental concepts. *IEEE Transactions on Power Delivery*, 2012.

[3] Wenxia Liu et.al. A bi-level interval robust optimization model for service restoration in flexible distribution networks. *IEEE Transactions on Power Systems*, 36(3):1843–1855, 2020.

[4] Shiwu Liao et.al. An improved two-stage optimization for network and load recovery during power system restoration. *Applied Energy*, 2019.

[5] Reza Roofegari Nejad and Wei Sun. Distributed load restoration in unbalanced active distribution systems. *IEEE Transactions on Smart Grid*, 10(5):5759–5769, 2019.

[6] Y Wang, A Oulis Rousis, D Qiu, and G Strbac. A stochastic distributed control approach for load restoration of networked microgrids with mobile energy storage systems. *International Journal of Electrical Power & Energy Systems*, 148:108999, 2023.

[7] Zain ul Abdeen et.al. Enhancing distribution system resilience: A first-order meta-rl algorithm for critical load restoration. In *2024 IEEE (SmartGridComm)*, pages 129–134. IEEE, 2024.

[8] Yiyun Yao et.al. Multi-agent reinforcement learning for distribution system critical load restoration. In *2023 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2023.

[9] Roshni Jacob et.al. Real-time outage management in active distribution networks using reinforcement learning over graphs. *Nature Communications*, 2024.

[10] Rui Fu, Zhiqi Xu, Beibei Wong, Hui Qian, Ling Ju, and Wei Jiang. Switch state identification in distribution network based on edge computing. In *2021 IEEE Sustainable Power and Energy Conference (iSPEC)*, pages 2318–2323. IEEE, 2021.

[11] Ryan Lowe et.al. Multi-agent actor-critic for mixed cooperative-competitive environments. *NeurIPS*, 30, 2017.

[12] Wenzhang Liu et.al. Xuance: A comprehensive and unified deep reinforcement learning library. *arXiv preprint arXiv:2312.16248*, 2023.