LARA: LLM-based Agile Power Distribution Network Restoration from Disastrous Events

Jishnu Warrier, Heqing Huang, Yuzhang Lin, Sai Qian Zhang
Tandon School of Engineering, New York University
{jishnu.warrier, heqinghuang, yuzhang.lin, sai.zhang}@nyu.edu

Abstract

Restoring power distribution networks after disruptions demands rapid, reliable coordination across repair crews, mobile power sources, and switching actions under strict constraints. Classical optimization yields high-quality plans but can be slow, while reinforcement learning often requires feeder-specific training and careful reward shaping. We recast restoration as language-conditioned planning: a large language model generates high-level restoration plans over a compact pre-validated catalogue of feasible actions. This constrained generation design makes decisions reliably, scalably, and interpretably, and allows for real-time human-in-the-loop decision-making while requiring no topology-specific setup or retraining. Our method achieves near-mixed-integer-linear programming performance on the IEEE 13-node standard power distribution feeder and outperforms a time-capped MILP solver on the IEEE 33-node standard feeder by around 13%, while using less than 1% of its wall-clock runtime.

1 Introduction

The timely restoration of power distribution networks after extreme weather events and natural disasters is critical to the energy resilience of modern societies (Shen et al., 2020). It is a complex sequential decision-making problem, requiring optimally coordinated actions including dispatching repair crews (RCs), routing and connecting mobile power sources (MPSs), and executing radiality-safe switch operations under tight resource and operational constraints. These actions determine which power loads are energized at each time step, and thus directly impact the scale and duration of power outages after disastrous events. In practice, multiple factors complicate the decision-making, e.g., RCs travel and repair over multiple time steps, MPSs move and supply limited power to isolated islands of the network, and only loop-free switch closures are legal; restoration is therefore a temporally coupled, constraint-laden combinatorial decision-making problem rather than a single-shot topology tweak (benchmarked on standard IEEE 13/33-node feeders) (Kersting, 2001; Baran and Wu, 1989a).

A large body of work frames restoration as mixed integer linear programming (MILP) optimization that jointly schedules repair-crew routing/repair, MPS placement/dispatch, and feeder reconfiguration under radiality and power limits (time-indexed over a horizon) (Lei, 2019; Lei et al., 2019; Ye et al., 2021; Ding et al., 2020; Yang et al., 2023; Hom, 2022; Wan, 2024; Kim and Dvorkin, 2019; Nazemi et al., 2021; Yao et al., 2020). MILP is the mainstream solution for power distribution system restoration because when computation completes without time constraints, the solution is guaranteed to be globally optimal (ignoring small approximation errors from the linearized power flow representation commonly used in distribution networks where voltages remain close to nominal values). This makes time-relaxed MILP an oracle reference for solution quality. However, achieving such optimality comes at the cost of very high computational burden, and no universal fast solver currently exists for problems of this complexity. Although MILP-based approaches provide theoretically optimal decisions, they remain challenging to deploy directly in real-world operational environments where utility field engineers must make rapid decisions based on experience. In our experiments, we use the MILP formulation of Lei (2019) as a reference on small systems. For larger systems we implement time-capped MILP (Ding et al., 2020) with up to one hour of computation time. While the 5-minute restoration timestep requires decisions within that window, we allow MILP one hour per decision, which is far beyond practical limits but can be viewed as close to an idealized solution.

To avoid repeated large-scale optimization problem solutions, recent work casts restoration as a cooperative partially observable Markov game and applies centralized-training, decentralized-execution (CTDE) multi-agent reinforcement learning (MARL). Popular choices include proximal policy optimization (PPO) based algorithm with a centralized critic for coordination (Sch, 2017; Yu2, 2021), and value-factorization methods such as Value Decomposition Networks and Q-Mixing Networks that decompose the joint action-value while enabling decentralized execution (Sun, 2018; Ras, 2018). Applied to restoration problems, CTDE agents coordinate RC/MPS actions under action masking and reward shaping, showing fast inference at test time but requiring substantial training steps and careful reward shaping, with transfer brittleness across feeders/fault patterns (Jacob et al., 2024; Wang et al., 2024). Hierarchical MARL mitigates long-horizon credit assignment by introducing macro-choices, yet still lacks hard feasibility guarantees without explicit guards.

The complementary limits of MILP (slow, exponentially scaling runtimes) and RL (training- and topology-specific policies with brittle transfer) motivate a third path: *language-conditioned planning over constrained affordances*. Two critical factors make large language models (LLMs) uniquely suited to this problem: computational scalability and adaptability to human-in-the-loop interventions.

First, LLMs provide fast inference that scales gracefully. While MILP solvers achieve near-optimal solutions on small networks in seconds, they encounter exponential complexity on larger topologies. In our experiments, time-capped MILP required up to 60 minutes per decision on the IEEE 33-node feeder, far exceeding the 5-minute decision window, whereas LLMs maintain sub-20-second latency.

Second, LLMs accept mid-episode natural language updates reflecting real-world volatility. During disaster response, operators receive fragmented field reports (road closures, revised damage estimates). Encoding these into MILP requires reformulating and cold-starting the optimization, while RL agents face out-of-distribution states. LLMs interpret natural language patches and immediately replan.

Prior work shows how language-conditioned affordances constrain LLM generation and offload symbolic checks (Ahn et al., 2022; Yao et al., 2023; Schick et al., 2023). To our knowledge, no prior LLM system addresses power distribution restoration with joint RC-MPS-switch coordination and explicit safety guarantees.

In this paper, we propose *L̲LM-based A̲gile Power Distribution Network R̲estoration from DisA̲strous Events* (LARA), which follows this affordance-first, language-conditioned planning paradigm. Our contributions are:

- We formulate power distribution network restoration as a *language-conditioned planning* problem, enabling LLMs to reason over constrained, prevalidated action spaces.
- LARA proposes an affordance-first, language-driven planning approach that decouples feasibility computation from LLM reasoning.
- LARA achieves near–MILP performance on IEEE-13 node and exceeds time-capped MILP by close to 13% on IEEE 33-node feeders with zero constraint violations using less than 1% of the wall-clock runtime of the MILP.

2 Methodology

In this section, we detail the LARA method and begin by formulating the problem in Section 2.1. Figure 1 provides a high-level workflow: (1) greedy switch reconfiguration handled by the environment (Sec. 2.2), (2) affordance precomputation exposing only topology-safe options (Sec. 2.3), (3) LLM synthesis of RC/MPS macro-actions over those IDs (Sec. 2.5), and (4) validate–execute–replan until any agent becomes idle; this loop is executed by the simulation environment. We also accept mid-episode human updates (Sec. 2.7).

2.1 Problem Formulation

We model post-disaster restoration as a discrete-time process $t = 0, \dots, T$ on a radial feeder with a single source and many load nodes. Each load n has active/reactive demand (P_n, Q_n) and a priority weight $W_n \in \{1, \dots, 10\}$. Each damage point (DP) k lies on a line and has remaining repair workload w_k. The controllable agents are RCs, MPSs, and switches. Each RC has a travel speed and a repair rate (work units per step); each MPS has a travel speed, maximum real/apparent power $(P_{\max}, S_{\max})$, and a finite energy budget. An *island* is a group of nodes that currently cannot receive power from the source because faults or open switches break the path. Let S denote the set of controllable switches and $S_{\text{open},t} \subseteq S$ the subset that is open at time t.

At time t, the state s_t includes agent posi-

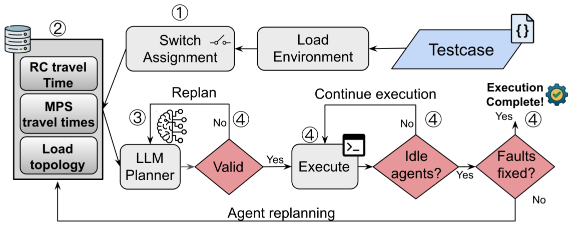

Figure 1: LARA architecture diagram

tions/resources, the open/closed switch set, $\{w_k\}$, and the energized loads L_t. A joint macro-action is

$$a_t = \left(\alpha_{\text{crew},t},\ \alpha_{\text{MPS},t},\ \alpha_{\text{switch},t}\right),$$

where $\alpha_{\text{crew},t}$ assigns each crew to a fault (or idle), $\alpha_{\text{MPS},t} = (m_t, v_t, C_t)$ selects an MPS m_t, a connection node v_t, and a feasible subset of loads C_t to supply, and $\alpha_{\text{switch},t} \subseteq S_{\text{open},t}$ is a set of switch closures. The simulator advances the state via $s_{t+1} = F(s_t, a_t)$, where F is a simulator transition function that updates the state, by applying agent travel, repair progress, switch toggles and MPS connection.

We maximize cumulative prioritized load restored:

$$\max_{\{a_t\}_{t=0}^{T}} \sum_{t=0}^{T} \sum_{n \in L_t} P_n W_n. \tag{1}$$

Hard constraints (enforced by the simulator). (i) *Radiality*: only legal (cycle-free) switch closures are allowed each step; the feeder remains loop-free at all times. (ii) *MPS power limits*: whenever an MPS supplies C_t, the real and apparent power satisfy $\sum_{n \in C_t} P_n \leq P_{\max}$ and $\sqrt{\left(\sum_{n \in C_t} P_n\right)^2 + \left(\sum_{n \in C_t} Q_n\right)^2} \leq S_{\max}$. (iii) *MPS energy budget*: each MPS has finite energy and cannot supply once depleted. (iv) *Kinematics/timing*: RCs and MPSs move at finite speeds; repairing fault k by a crew with repair rate r takes time w_k/r; a switch closure consumes one step; an MPS requires travel plus a one-step connection before supplying. All feasibility checks are applied before actions execute, and F respects these dynamics when producing s_{t+1}.

The constraints above directly govern the scheduled entities (RCs, MPSs, switches) and are strictly enforced by our simulator. In practice, distribution systems also require power-flow-driven constraints such as nodal voltage limits and line thermal capacity limits. For distribution networks operating near nominal voltage, LinDistFlow provides a widely accepted linearized approach (Baran and Wu, 1989b; Taheri et al., 2025; Lei, 2019; Sun et al., 2023). Incorporating such constraints into our affordance precomputation is straightforward, as they impose additional linear inequalities on feasible load subsets, but we leave this to future work.

2.2 Greedy Switch Reconfiguration

Switch selection is performed by the simulator using a fast greedy procedure that avoids the $2^{|\mathcal{S}|}$ combinatorial search an LLM cannot reliably execute. A reconfiguration is triggered whenever a reconfiguration is required prior to the LLM call. Starting from the current open/closed set, we enforce legality by preserving radiality via a reachability/loop check. For each still-open legal switch s, we compute its marginal benefit $\Delta(s)$, the newly energized weighted load if s were closed using a Breadth-first search from source node with island accounting, which is $O(|\mathcal{N}|)$. We then close the switch with the largest $\Delta(s) > 0$, update the graph, recompute all $\Delta(\cdot)$ values, and repeat until no positive-gain switch remains. This yields $O(|\mathcal{S}| \cdot |\mathcal{N}|)$ work per reconfiguration event which remains a modest computational cost even on large feeders.

2.3 Precomputation

From the current state we build compact tables (IDs with vetted numbers): (i) RC to DP: for each crew-fault pair, travel time, single/paired completion times (T_1, T_2), and a time-weighted benefit proxy; (ii) MPS to Island: for each island, nearest valid node and feasible load-combinations that satisfy $P_{\max}$ and $S_{\max}$, each with travel/connect time and benefit. All feasibility checks (radiality, MPS limits, DP capacity) are enforced here.

2.4 Interface structure

The LLM receives a structured natural language description of the current state comprising: (i) network topology with node connectivity and distances, (ii) agent states (RC/MPS locations, resources, and status), (iii) damage point locations and remaining workloads, (iv) island specifications (isolated node sets with aggregate power demand and priority), (v) precomputed affordance tables (RC-to-fault and MPS-to-island options with travel times, repair durations, and benefit estimates), and (vi) operative constraints (radiality, MPS power limits, energy budgets). The LLM outputs a structured plan with: (i) crew assignments mapping each RC to a target fault or idle status (e.g., RC1→DP2), (ii) MPS assignments specifying connection node and load subset (e.g., MPS1→node 5, supply {node 5,node 6}), and (iii) natural language reasoning traces explaining the coordination strategy. The simulator validates all returned IDs against the precomputed tables before execution.

2.5 LLM Plan Synthesis

To improve robustness, we sample $k \in \{1, 3, 5\}$ diverse plans in parallel and deduplicate them. While the LLM could potentially provide infeasible actions, we have not encountered this due to careful prompting that requires the LLM to choose from the precomputed set of feasible actions.

2.6 Simulation-Guided Selection

Each RC/MPS candidate proposed by the LLM is evaluated through a short rollout that follows the same physical model used in the precomputation phase, along with the greedy switching module described above. The simulation proceeds until the first reassignment trigger occurs, such as when an agent becomes idle or a planned action is completed. A short-horizon restoration score consistent with Eq (1) is then computed, and highest scoring candidate is selected, with ties resolved in favor of earlier proposals. The chosen candidate is executed step by step until an agent becomes idle, at which point a reassignment request is issued to the LLM.

2.7 Human-in-the-Loop Decision-Making

The planner allows real-time edits by human operators during post-disaster response. At any step, operators can modify crew speeds, fault durations, MPS limits, or block/clear roads reflecting updates common in emergency scenarios. We apply a state patch, recompute only affected affordances, and replan using LLM. Unlike MILP, this incurs no retraining or full re-solving overhead.

3 Experiments

We evaluate LARA on two standard distribution feeders: the IEEE 13-node test feeder, with 50 randomized scenarios, and the IEEE 33-node test feeder, with 10 representative scenarios. Each scenario defines fault locations and capacities, initial positions and speeds of RCs and MPSs, initial switch configurations, and load parameters (P, Q) with priority weights $W \in 1, \ldots, 10$. Detailed descriptions of both test feeders, including modifications and scenario generation procedures, are provided in Appendix B.

For comparison, we implement the MILP following the approach of Lei (2019). This implementation serves as a reference baseline with a total inference time capped at one hour. We also adopt a multi-agent PPO baseline that employs CTDE agents, incorporating action masking to filter infeasible operations. This baseline is referred to as *CTDE-RL*. Training (refer to A.1) is conducted separately on the IEEE 13-node and IEEE 33-node test feeders using a single NVIDIA A100 GPU for 500k environment steps per feeder. The final checkpoints are selected based on validation rewards, and inference is performed on the CPU. We also implement a Greedy Proximity heuristic as a fast classical baseline with computational budget similar to LARA. This method assigns each RC to the nearest unrepaired fault, connects each MPS to the nearest feasible island node with maximum weighted load within power limits, and applies the same greedy switch reconfiguration as LARA. It requires minimal computation (<1s) but lacks global coordination and long-horizon reasoning. Our planner is primarily executed on LLaMA-3.3-70B, with additional evaluations conducted on Gemini 2.5 Pro and Claude 3.7 Sonnet (Thinking). We perform ablation studies using $k \in \{1, 3, 5\}$ and temperature values $T \in \{0.2, 0.5, 0.8\}$ to assess the impact of proposal diversity.

3.1 Evaluation Metrics

Our primary evaluation metric is %MILP, which measures relative performance: for each scenario, we compute $(100 \times E_{\text{method}}/E_{\text{MILP}})$ where E denotes cumulative weighted power restored, then average across scenarios; higher is better. We also report two secondary metrics: restoration timesteps, which counts the total simulator steps required to fully energize all loads (lower indicates faster completion), and decision latency, which reports the median wall-clock time required by each method to compute actions at a single decision point (this must remain below the 5-minute restoration timestep duration to avoid delaying execution in real deployments).

3.2 Main Results

Table 1 summarizes LARA against a time-capped MILP reference and a CTDE-RL baseline on the IEEE 13-node and 33-node feeders. On the IEEE 33-node feeder, the time-capped MILP solver reached optimality in only 1 out of 10 scenarios within the one hour time limit, confirming the com-

Feeder	Method	%MILP (mean)	Restoration Timesteps (mean)	Latency (median)
IEEE-13 node	**LARA** (k=3)	**95.58%**	**8.27**	**5.2s**
IEEE-13 node	RL (CTDE)	94.45%	6.51	0.5s
IEEE-13 node	MILP	100.0%	4.44	5s
IEEE-13 node	Greedy Proximity	58.20%	12.9	<1 s
IEEE-33 node	**LARA** (k=3)	**112.99%**	**12.1**	**18.1s**
IEEE-33 node	RL (CTDE)	75.16%	22.5	0.5s
IEEE-33 node	MILP (time capped)	100.0%	14.6	58.8mins
IEEE-33 node	Greedy Proximity	15.00%	29.6	<1 s

Table 1: Performance comparison of methods across IEEE-13-node and IEEE-33-node feeders.

Model (k=3)	IEEE-13 node (%MILP)	IEEE-33 node (%MILP)	Latency (s)
LLama 3.3 70B	95.58%	**112.99%**	**18.1**
Claude 3.7 Sonnet	95.99%	103.27%	120.6
Gemini 2.5 Pro	**97.85%**	96.80%	100.4

Table 2: Backbone sensitivity and latency at k=3 across LLMs (per-feeder mean score; higher is better).

putational intractability of exact optimization on this scale. The remaining scenarios terminate at the time cap with suboptimal but feasible solutions. On the smaller IEEE 13-node feeder, MILP consistently reaches optimality well within the cap, making it a reliable oracle baseline for that benchmark.

LARA achieves 112.99% of MILP on IEEE-33 node, exceeding the time-capped baseline. LARA achieves this in 12.1 steps on average, fewer than time-capped MILP (14.6) and CTDE-RL (22.5), indicating more efficient crew/MPS coordination. On the IEEE-13-node testcases, LARA attains 95.6% of MILP while averaging 8.27 steps vs. MILP's 4.44 and RL's 6.51. This is expected as our objective maximizes Eq 1 (weighted load-timesteps), which rewards early restoration of high-priority loads rather than pure makespan. LARA restores critical loads earlier, yielding higher cumulative reward despite longer completion time.

LARA's performance gain on the IEEE 33-node test system (112.99% of MILP) relative to the IEEE 13-node test system (95.58%) demonstrates its scalability. The model effectively navigates larger search spaces where classical solvers become computationally bottlenecked.

To demonstrate portability of the interface, we run the base case with k=3 on three backbones (LLama 3.3 70B, Gemini 2.5 Pro, Claude 3.7 Sonnet (Thinking); the comparison is provided in Table 2. The models perform comparably and achieve near 95% on the IEEE 13-node test feeder, however, on the IEEE 33-node test feeder LLama3.3 70b clearly outperforms Claude and Gemini.

Setting (k, T)	IEEE-13 node(% MILP)	IEEE-33 node(% MILP)
(3, 0.2)	95.58	112.99
(3, 0.5)	94.52	110.32
(3, 0.8)	91.24	99.93
(1, 0.2)	95.26	103.83
(3, 0.2)	95.58	112.99
(5, 0.2)	96.07	112.67

Table 3: Ablation study. First 3 rows vary temperature T with fixed k=3; last three vary k with fixed T=0.2

3.3 Ablations

We study two knobs of the method: (i) number of proposals $k \in \{1, 3, 5\}$ shows quality increases with diminishing returns; Increasing k from 1 to 3 provides a modest benefit in IEEE-13 node and a sizable improvement in IEEE-33 node test feeder, however as we increase k further to 5 there does not seem to be an increase in IEEE 33-node and a minor increase in IEEE 13-node test feeder, this is because the LLM tends to give fairly similar results on different LLM calls (since we are testing with T=0.2), the minor difference in IEEE 33-node test feeder performance can be attributed to selection of different set of MPS nodes in k=3 and k=5. (ii) testing temperature values 0.2, 0.5, 0.8 at k=3 shows marginal drop in the model performance between T=0.2 and T=0.5 and a sharp drop when we increase T to 0.8. Our best model performance was achieved with k = 3 and temperature of 0.2, balancing performance and computational cost.

4 Conclusion

We cast power distribution system restoration as language-conditioned planning over prevalidated affordances, separating hard feasibility from LLM reasoning. We compose joint RC-MPS macro-actions that are safe by construction and scored under strict constraints. Results show that LARA approaches MILP quality on small feeders (IEEE 13-node) and surpasses time-capped MILP and CTDE-RL on larger benchmarks (IEEE 33-node) with zero constraint violations and sub-20-second decision latency.

While real-world distribution networks can contain thousands of nodes, the number of dispatchable agents remains limited, typically fewer than ten per service region, which prevents combinatorial explosion. This makes LARA's design inherently scalable and generalizable: the system computes feasible affordance tables dynamically at runtime, requiring no topology-specific customization or fine-tuning.

Limitations

Our evaluation focuses on IEEE 13-node and IEEE 33-node test feeders with 3 to 4 agents per scenario. While these are standard benchmarks in power systems research, empirical validation on utility-scale feeders with hundreds of nodes would further demonstrate scalability. The modeled constraints focus on agent dispatch, radiality, and MPS power limits. Incorporating nodal voltage limits and line thermal capacity constraints via LinDistFlow remains future work. All results are simulation-based, and real-world deployment would require validation against operator practices, communication delays, and field conditions.

References

2017. Proximal policy optimization algorithms. In *ICML Workshop / arXiv preprint arXiv:1707.06347*.

2018. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *ICML*.

2018. Value-decomposition networks for cooperative multi-agent learning. In *AAMAS (Extended Abstract) / arXiv:1706.05296*.

2019. Resilient disaster recovery logistics of distribution systems: Co-optimizing service restoration with repair crew and mobile power source dispatch. *IEEE Transactions on Power Systems*.

2021. The surprising effectiveness of MAPPO in cooperative multi-agent games. In *NeurIPS Deep RL Workshop / arXiv:2103.01955*.

2022. Improvement of distribution systems' resilience via operational resources and demand response. *Electric Power Systems Research*.

2024. Multiperiod distribution system restoration with routing: Repair crews, mobile electric vehicles, and networked microgrids. *IEEE Transactions on Power Systems*.

Michael Ahn, Anthony Brohan, Noah Brown, and 1 others. 2022. Do as i can, not as i say: Grounding language in robotic affordances. In *Robotics: Science and Systems (RSS)*.

M. E. Baran and F. F. Wu. 1989a. Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Transactions on Power Delivery*, 4(2):1401–1407.

M. E. Baran and F. F. Wu. 1989b. Optimal capacitor placement on radial distribution systems. *IEEE Transactions on Power Delivery*, 4(1):725–734.

B. Chen, C. Chen, J. Wang, and K. L. Butler-Purry. 2018. Multi-time step service restoration for advanced distribution systems and microgrids. *IEEE Transactions on Smart Grid*, 9(6):6793–6805.

Tao Ding, Zhaoyu Wang, Wei Jia, Bo Chen, Chen Chen, and Mohammad Shahidehpour. 2020. Multiperiod distribution system restoration with routing repair crews, mobile electric vehicles, and soft-open-point networked microgrids. *IEEE Transactions on Smart Grid*, 11(6):4795–4808.

Roshni Anna Jacob, Steve Paul, Souma Chowdhury, Yulia R. Gel, and Jie Zhang. 2024. Real-time outage management in active distribution networks using reinforcement learning over graphs. *IEEE Transactions on Smart Grid*.

William H. Kersting. 2001. Radial distribution test feeders. In *IEEE Power Engineering Society Winter Meeting*.

Jinsub Kim and Yury Dvorkin. 2019. Enhancing distribution system resilience with mobile energy storage and microgrids. *IEEE Transactions on Smart Grid*, 10(5):4996–5006.

S. Konar, A. K. Srivastava, and A. Dubey. 2023. Distributed optimization for autonomous restoration in der-rich distribution system. *IEEE Transactions on Power Delivery*, 38(5):3205–3217.

Shunbo Lei, Chen Chen, Hui Zhou, and Yunhe Hou. 2019. Routing and scheduling of mobile power sources for distribution system resilience enhancement. *IEEE Transactions on Smart Grid*, 10(5).

Meysam Nazemi, Payman Dehghanian, Xiaonan Lu, and Chen Chen. 2021. Uncertainty-aware deployment of mobile energy storage systems for distribution grid resilience. *IEEE Transactions on Smart Grid*, 12(4):3200–3214.

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

R. Roofegari nejad, W. Sun, and A. Golshani. 2019. Distributed restoration for integrated transmission and distribution systems with ders. *IEEE Transactions on Power Systems*, 34(6):4964–4973.

Timo Schick, Jane Dwivedi-Yu, and 1 others. 2023. Toolformer: Language models can teach themselves to use tools. arXiv:2302.04761.

Feifan Shen, Qiuwei Wu, and Yusheng Xue. 2020. Review of service restoration for distribution networks. *Journal of Modern Power Systems and Clean Energy*, 8(1):1–14.

X. Sun, J. Chen, H. Zhao, W. Zhang, and Y. Zhang. 2023. Sequential disaster recovery strategy for resilient distribution network based on cyber–physical collaborative optimization. *IEEE Transactions on Smart Grid*, 14(2):1173–1187.

B. Taheri, R. K. Gupta, and D. K. Molzahn. 2025. Optimizing parameters of the lindistflow power flow approximation for distribution systems. *IEEE Transactions on Smart Grid*.

L. Vu, T. Vu, T. L. Vu, and A. Srivastava. 2024. Multi-agent deep reinforcement learning for distributed load restoration. *IEEE Transactions on Smart Grid*, 15(2):1749–1760.

Y. Wang and 1 others. 2019. Coordinating multiple sources for service restoration to enhance resilience of distribution systems. *IEEE Transactions on Smart Grid*, 10(5):5781–5793.

Yi Wang, Dawei Qiu, Fei Teng, and Goran Strbac. 2024. Towards microgrid resilience enhancement via mobile power sources and repair crews: A multi-agent reinforcement learning approach. *Applied Energy*.

Xiaodong Yang, Chongbo Xu, Jinyu Wen, Youbing Zhang, Qiuwei Wu, Wenping Zuo, and Shijie Cheng. 2023. Cooperative repair scheduling and service restoration for distribution systems with soft open points. *IEEE Transactions on Smart Grid*, 14(3):1827–1841.

Shunyu Yao, Jeffrey Zhao, and 1 others. 2023. React: Synergizing reasoning and acting in language models. arXiv:2210.03629.

Shuyu Yao, Peng Wang, Xiaoyu Liu, Hong Zhang, and Tianshu Zhao. 2020. Rolling optimization of mobile energy storage fleets for resilient service restoration. *IEEE Transactions on Smart Grid*, 11(2):1030–1043.

Zhigang Ye, Chen Chen, Bo Chen, and Kai Wu. 2021. Resilient service restoration for unbalanced distribution systems with distributed energy resources by leveraging mobile generators. *IEEE Transactions on Industrial Informatics*, 17(2):1386–1396.

A Appendix

A.1 CTDE-RL Baseline Details

We implement a centralized-training, decentralized-execution (CTDE) PPO baseline ("CTDE-RL"). A centralized critic consumes a compact global state embedding, while decentralized actors—one each for RC, MPS, and switches produce actions for their respective agents. Illegal actions are masked before sampling to ensure feasibility.

For comparability with LARA, the same simulator and physics are used. The environment enforces radiality at every step and hard-checks MPS real/apparent-power limits. Episodes terminate when all faults are repaired or when a fixed step cap is reached, identical to the setting for the other methods.

Action spaces are discrete. The RC actor selects a crew-to-fault assignment or an idle/continue choice; faults that are unreachable or already completed are masked. The MPS actor selects a connection node together with an index to a feasible load subset; options violating $P_{\max}$ or $S_{\max}$ are masked. Unlike LARA, switching is learned: a dedicated switch actor chooses a legal switch toggle (or no-op) among currently open, cycle-free candidates, with illegal closures masked by the environment.

Observations are factorized by agent and normalized online. Each RC observes its position, speed, repair rate, current assignment, distances to a shortlist of feasible faults, and the remaining work for that shortlist. Each MPS observes its position, travel speed, $(P_{\max}, S_{\max})$, remaining energy, and a shortlist of feasible connection options (node identifier plus a summary of the load subset). The critic additionally receives global summaries of all damage points (locations and remaining work), a summary of energized load, and a compact reachability fingerprint induced by the current switch and fault configuration.

The per-step reward equals the newly energized weighted load $\sum_{n\in\Delta L_t} P_n W_n$, i.e., the exact increment of the reward objective; no additional shaping is used. Networks are multilayer perceptrons with hidden widths 256, 128, and 64 and ReLU activations; hidden layers use layer normalization, and categorical identifiers (e.g., fault or connection indices) are embedded and concatenated to continuous features.

Optimization uses PPO with the clipped surrogate objective and Adam. Unless otherwise noted, $\gamma = 0.99$, $\lambda = 0.95$, clip ratio 0.2, learning rate 3×10^{-4}, entropy coefficient 0.01, value-loss coefficient 0.5, gradient-norm clip 0.5, rollout length 128, four minibatches, and four epochs per update, with N parallel environments chosen to keep the effective batch size stable. We train separately on the IEEE-13 node and IEEE-33 node test feeders for 500k environment steps each on a single NVIDIA A100 GPU, select checkpoints by validation reward, and evaluate on CPU with masked greedy (argmax) action selection.

Separate training per feeder is required because the observation and action cardinalities depend on feeder topology and scale (numbers of nodes, switches, feasible connection options, and masking sets). A single shared policy would face mismatched heads and distribution shift across feeders, which empirically degrades transfer.

Scenario splits are identical across methods: CTDE-RL uses training and validation scenarios to fit and select checkpoints, and all methods are finally evaluated on the same held-out test scenarios reported in Section 3.2. The underlying constraints and physics are identical; only the switching decision differs, being learned in CTDE-RL and greedy in LARA.

B Sample Topologies

B.1 IEEE 13-Node Test Feeder

The IEEE 13-node feeder is a widely adopted benchmark for distribution system research (Kersting, 2001). Its complete technical specifications are published and openly available from the IEEE PES Test Feeders repository.[1] Although relatively small, it captures typical characteristics of real distribution grids, including unbalanced load distribution, diverse power factor conditions, and radial topology under normal operation, which makes it well-suited for proof of concept studies. This benchmark and its variants are commonly used in top-tier power systems research to validate new restoration methods (Roofegari nejad et al., 2019; Vu et al., 2024; Wang et al., 2019; Chen et al., 2018).

In our study, we make one minor modification to the standard IEEE 13-node feeder: we add a normally-open tie line to enable partial load restoration when a fault occurs. All other electrical parameters (line impedances, load values, voltage levels) remain identical to the publicly available IEEE data. This adjustment increases the solution space and better demonstrates the capability of the proposed method. Since the IEEE model does not provide assumptions regarding transportation distances between nodes, we reasonably derive these distances from line impedances and convert them into integer per-unit representations. Such small adaptations to IEEE benchmark feeders for evaluating new approaches are widely accepted practice in power system studies (Roofegari Nejad and Sun, 2019; Konar et al., 2023).

We generate 50 randomized scenarios by sampling 2–5 simultaneous faults (locations and workloads), initial positions and speeds for 2 repair crews and 1 MPS, and load priority weights $W \in \{1, \ldots, 10\}$. Agent kinematics follow finite speeds: RCs move and repair at fixed rates per timestep, while each MPS travels then spends one connection step before supplying feasible load subsets subject to its $P_{\max}$, $S_{\max}$, and finite energy budget. Switch actions are determined by the greedy reconfiguration module and always preserve radiality. Episodes terminate when all faults are repaired and no island remains unserved. The network diagram is shown in Figure 3.

B.2 IEEE 33-Node Test Feeder

We use the standard radial 33-node feeder (Baran and Wu, 1989b), which is also publicly available from the IEEE PES Test Feeders repository. This larger benchmark provides a more challenging test case to verify generalizability and scalability. The feeder operates with a single source node and includes priority-weighted loads ($W \in \{1, \ldots, 10\}$) representing varying criticality levels across the distribution network.

Scenarios contain 6 to 8 simultaneous faults, 2 repair crews (RCs), and 2 mobile power sources (MPSs). As with the 13-node feeder, RCs move at finite speed and repair at a fixed per-step rate. Each MPS travels, then spends one connection step and may supply any feasible subset of local loads subject to its $P_{\max}$, $S_{\max}$, and finite energy budget. Switch actions are set by the greedy reconfiguration module and always preserve radiality. We evaluate 10 representative scenarios with randomized fault placements and workloads, initial RC and MPS positions and speeds, and load priorities. Episodes terminate when all faults are repaired and no island remains without grid or MPS supply. As shown in the simulation results, the restoration problem in the IEEE 33-node feeder becomes highly complex, as the MILP solver can hardly find the optimal solution within one hour. The network diagram is shown in Figure 2.

[1] https://cmte.ieee.org/pes-testfeeders/resources/

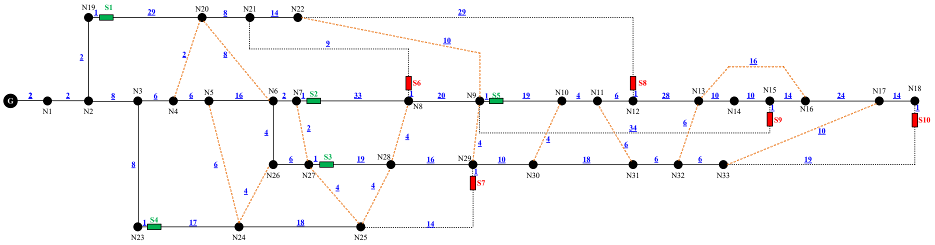

Figure 2: IEEE 33-node test feeder diagram (S1–S10 are switches).

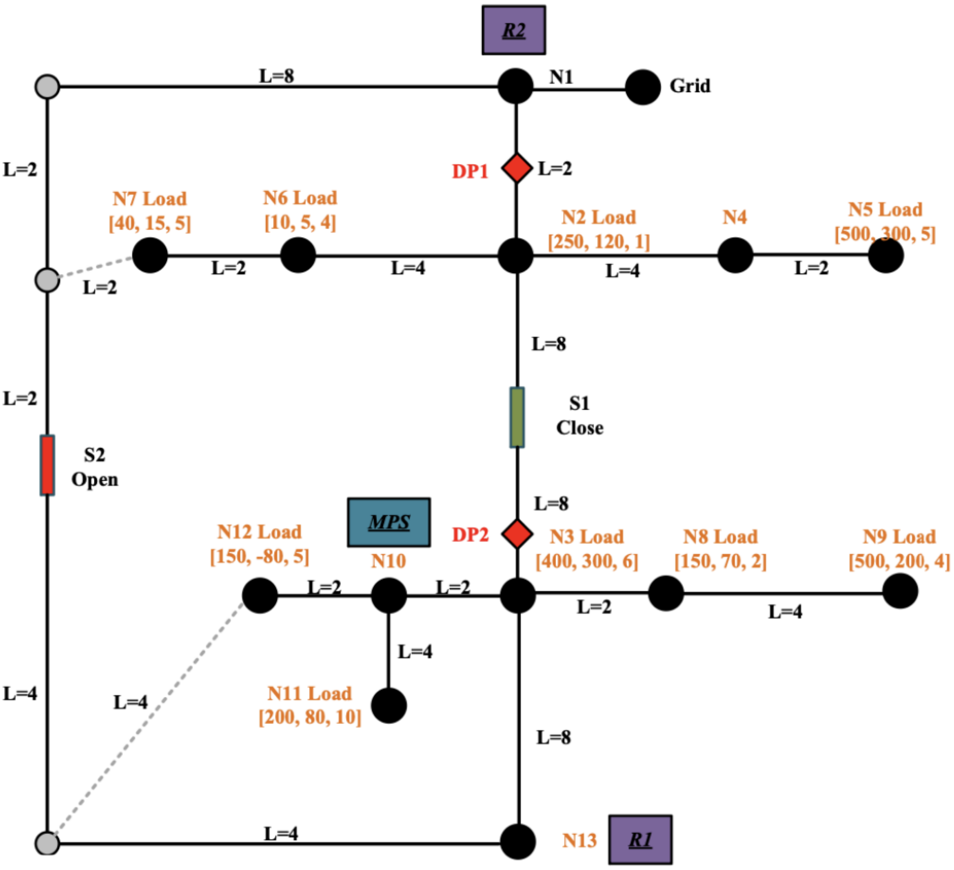

Figure 3: Sample IEEE 13-node test feeder test case. N1–N13 are nodes; R1, R2 are RCs; S1, S2 are switches; DP1, DP2 are faults. Node labels show [active power, reactive power, weight].