

Explainable graph reinforcement learning for resilience assessment and self-healing decision-making in distribution networks

YeHaoXiang Liu

liugehaoxiang@outlook.com

Chizhen Wang

Research Article

Keywords:

Posted Date: October 20th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10774377/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Explainable graph reinforcement learning for resilience assessment and self-healing decision-making in distribution networks

Yehaoxiang Liu^{1*} and Chizhen Wang²

^{1*}School of Big Data and Information Engineering, Xinjiang University
of Technology, Hotan, 848000, P.R. China.

²College of Animal Science, Xinjiang Agricultural University, Urumqi,
830000, P.R. China.

*Corresponding author(s). E-mail(s): liuyehaoxiang@outlook.com;
Contributing authors: 2390340965@qq.com;

Abstract

The rapid integration of distributed generation and the growing frequency of extreme weather events are stressing the security of distribution-network supply, and fault self-healing must now reconcile three competing demands: decision quality, real-time performance, and explainability. Mathematical programming yields near-optimal solutions but is too slow for online use, whereas existing deep reinforcement learning (DRL) methods decide within milliseconds but remain opaque, which limits operator trust. We present an explainable graph reinforcement learning method for resilience assessment and self-healing decision-making in distribution networks. The network is first modelled as a graph and a graph attention network (GAT) extracts topology-aware state representations; the GAT is then embedded as the shared backbone of a soft actor-critic (SAC) agent, whose maximum-entropy objective improves exploration in complex fault scenarios. A dual-track explainability mechanism combines intrinsic multi-head attention heatmaps with post-hoc SHAP attributions, providing evidence of both where the agent attends and why it decides, and a resilience index is embedded directly in the reward so that maximizing resilience becomes the training objective. Joint MATLAB-Python simulations on the IEEE 33-bus and 123-bus systems show that the method achieves decision quality within 3% of the mixed-integer linear programming optimum, reaches millisecond-level online decision times (roughly three orders of magnitude faster than conventional optimization), and produces interpretable attention maps and SHAP rankings.

Keywords: Explainable reinforcement learning, Graph attention network, Distribution network resilience, Self-healing control, Soft actor–critic, Co-simulation

1 Introduction

The large-scale integration of distributed generation (DG), together with increasingly frequent extreme weather events, has amplified both source- and load-side uncertainty in distribution networks and raised the risk of large-scale outages [1, 2]. Distribution-network resilience—the ability of a system to withstand, adapt to, and rapidly recover its power-supply function following an extreme disturbance—has therefore become a central concern [3, 4]. Resilience differs from classical reliability in that it emphasizes not only preventive capability before a fault but also the depth of performance degradation and the speed of recovery after it [5].

Fault self-healing is the core technical means of enhancing resilience. It restores as much critical load as possible, as quickly as possible, by operating tie and sectionalizing switches and by dispatching DG after a fault [6, 7]. Modern distribution networks are large, topologically complex, and rich in decision variables, so self-healing is fundamentally a high-dimensional, non-convex, strongly constrained combinatorial optimization problem that simultaneously demands decision quality, real-time performance, and explainability. Conventional methods rarely satisfy all three within a finite computation budget, which motivates new technical routes.

Resilience assessment has been studied extensively. Deterministic approaches quantify resilience as the area under the system performance curve [3, 4], while probabilistic extensions account for the uncertainty of faults and renewable output [8, 9]. These efforts provide the theoretical tools for resilience quantification but largely separate assessment from decision-making: resilience indices are seldom embedded directly into real-time control objectives.

For fault restoration, mathematical programming—including mixed-integer linear and mixed-integer quadratic programming—yields theoretically optimal solutions and remains the classical paradigm for large-scale restoration [10–12]. Its solution time, however, grows sharply with system size and cannot meet second- or millisecond-level online requirements. Heuristic algorithms such as particle swarm optimization and genetic algorithms reduce the computational burden but offer no guarantee on optimality or stability [13].

Deep reinforcement learning (DRL), which trains offline and decides online, offers an alternative. Deep Q-networks, deep deterministic policy gradient, and soft actor–critic (SAC) have been applied to voltage control, economic dispatch, and fault restoration [14–22], demonstrating millisecond-level decision capability. Conventional DRL, however, processes network data through fully connected or convolutional networks, which cannot explicitly exploit grid topology, and its decisions are opaque, which limits operator trust [23, 24].

Graph neural networks (GNNs) learn representations directly on non-Euclidean graph structures and therefore match the topology of distribution networks. Graph

attention networks (GATs), in particular, assign adaptive importance to neighbouring nodes through attention and have performed well in power-system state estimation and transient-stability assessment [25–27]. Combining GNNs with DRL to endow the policy network with topology awareness is an active research direction [28–31].

In explainable artificial intelligence (XAI), SHAP attributes model outputs to input features through game-theoretic Shapley values, LIME explains individual decisions by local linear approximation, and attention mechanisms expose importance distributions inside the model [32–35]. XAI has been introduced into power systems for load forecasting and fault diagnosis [36–38], but its application to self-healing—where safety and interpretability are paramount—remains nascent.

Three gaps persist. First, resilience assessment and self-healing decision-making are decoupled: resilience indices are not part of the real-time control objective, so assessment and decision diverge. Second, DRL policy networks are largely black boxes that ignore grid topology, which limits decision quality and generalization. Third, the self-healing decision process is opaque, so operators cannot understand or trust the agent’s rationale.

We address these gaps with an explainable graph reinforcement learning method, whose technical roadmap is shown in Fig. 1. The contributions are threefold. First, we fuse GAT with SAC by using the GAT as the shared backbone of the actor and critic, so the agent adaptively extracts topology-aware state representations and preserves high-quality restoration decisions in complex fault scenarios. Second, we build a dual-track explainability mechanism: intrinsic multi-head attention heatmaps reveal *where* the agent attends, and post-hoc SHAP values quantify the marginal contribution of each state feature to explain *why* it decides, giving dispatchers two mutually corroborating forms of evidence. Third, we embed a resilience index directly into the reward function, making resilience maximization the training objective and closing the loop between assessment and decision.

2 Problem formulation

2.1 Resilience definition and quantification

We characterize the dynamic evolution of the power-supply capability of a distribution network throughout a disturbance by a performance curve $F(t)$, defined as the ratio of served load to total load:

$$F(t) = \frac{P_{\text{served}}(t)}{P_{\text{total}}} \quad (1)$$

where $P_{\text{served}}(t)$ is the load power restored at time t and P_{total} is the total load. On this basis, the resilience index R is defined as the ratio of the area under the performance curve between fault occurrence and full restoration to the ideal performance area [3, 4]:

$$R = \frac{\int_{t_0}^{t_1} F(t) dt}{F_0 (t_1 - t_0)} \quad (2)$$

where t_0 and t_1 denote the instants of fault occurrence and full restoration, respectively, and F_0 is the pre-fault performance baseline. The index satisfies $R \in [0, 1]$; a

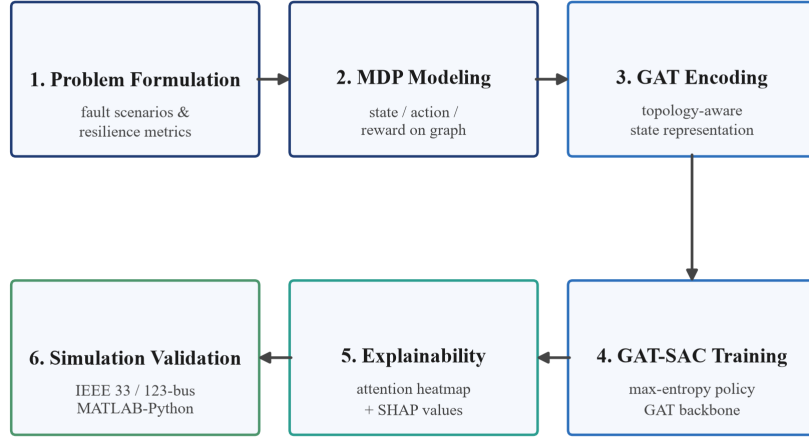


Fig. 1 Research technical roadmap of the proposed explainable graph reinforcement learning method

larger R indicates a stronger capability to maintain and restore supply under disturbance. Table 1 lists auxiliary indicators—load restoration rate, mean restoration time, and voltage deviation—that characterize resilience from complementary dimensions.

Table 1 Resilience quantification indicators for distribution networks

Indicator	Symbol	Definition	Meaning
Resilience index	R	$\int F(t) dt / (F_0 \Delta t)$	Composite measure of resilience
Load restoration rate	η_r	$P_{\text{restored}} / P_{\text{loss}}$	Fraction of lost load restored
Mean restoration time	T_r	$\frac{1}{N} \sum_i t_{r,i}$	Average restoration speed
Voltage deviation	δ_V	$\max_i V_i - 1 $	Degree of voltage-quality violation

The performance curve decomposes into three phases. In the withstanding phase the system sustains performance near F_0 through its own robustness; after the fault, performance drops rapidly to a minimum $F_{\min}$ as the system enters the adaptation phase in which self-healing takes effect; and in the recovery phase performance climbs back toward F_{rec} . The resilience index R measures the area under the curve across these phases, so a larger R corresponds to stronger withstanding capability, shallower degradation, and faster recovery. This framing provides the quantitative basis for embedding the resilience index into the reinforcement-learning reward.

2.2 Fault scenario modelling

We consider two typical fault scenarios: single-branch and multi-branch faults. After a fault, the sectionalizing switches on both sides of the faulted branch open to isolate

it, cutting off the downstream loads; self-healing then closes tie switches to reconfigure the network and dispatches DG to transfer the de-energized loads to alternative supply paths. For the IEEE 33-bus system, a fault scenario is abstracted as a combination of a faulted-branch set and an operable-switch set; concrete settings are given in Section 5.

In the single-branch scenario, the two sectionalizing switches adjacent to the faulted branch isolate the faulted section, the downstream loads lose supply, and self-healing restores them through a suitable tie switch. In the multi-branch scenario, several faults occur simultaneously; the de-energized region is larger and the choices of tie switches are coupled, so the decision space is more complex. We construct a single-branch fault (opening branch 12–13) and a double-branch fault (opening branches 2–19 and 6–26 simultaneously) to test the algorithm at different complexity levels.

2.3 Self-healing decision problem

The objective of self-healing is to maximize the restored load while accounting for losses and switching cost, subject to operational constraints:

$$\max \omega_1 \sum_{i \in \mathcal{B}} P_i^{\text{rest}} - \omega_2 P_{\text{loss}} - \omega_3 N_{\text{sw}} \quad (3)$$

subject to power-flow constraints, node-voltage bounds, branch-capacity limits, and radiality:

$$V_{\min} \leq V_i \leq V_{\max}, \quad \forall i \in \mathcal{B} \quad (4)$$

$$|I_{ij}| \leq I_{ij}^{\max}, \quad \forall (i, j) \in \mathcal{E} \quad (5)$$

$$g(\mathcal{S}, \mathcal{O}) = \text{Radial} \quad (6)$$

where $\mathcal{B}$ and $\mathcal{E}$ are the node and branch sets, P_i^{rest} is the load restored at node i , P_{loss} is the network loss, N_{sw} is the number of switching operations, and ω_1 – ω_3 weight the objectives. Eq. (4) enforces voltage bounds, Eq. (5) branch capacity, and $g(\cdot)$ in Eq. (6) imposes the radiality constraint determined by the sectionalizing-switch states $\mathcal{S}$ and tie-switch states $\mathcal{O}$.

In Eq. (3), the first term prioritizes load restoration and ω_1 is set large to reflect supply priority; the second term minimizes losses to guide the agent toward topologies with better power-flow paths; and the third term minimizes switching operations to reduce equipment wear and switching transients. The three weighted terms capture the trade-off among supply reliability, operating economy, and equipment lifetime. The radiality constraint $g(\cdot)$ is the central difficulty of reconfiguration: the restored network must remain radial and island-free, which can be verified by tree detection or by the rank condition of the bus–branch incidence matrix.

The problem is a mixed-integer nonlinear combinatorial optimization whose complexity grows sharply with network size and switch count. Mathematical programming solves it well offline but faces a computation bottleneck online. We therefore reformulate it as a Markov decision process (MDP) and solve it with graph reinforcement learning for fast online decisions.

2.4 Markov decision process formulation

An MDP comprises a state space $\mathcal{S}$, an action space $\mathcal{A}$, a transition probability $\mathcal{P}$, and a reward function $\mathcal{R}$ (Fig. 2). We model self-healing as an MDP with the following elements.

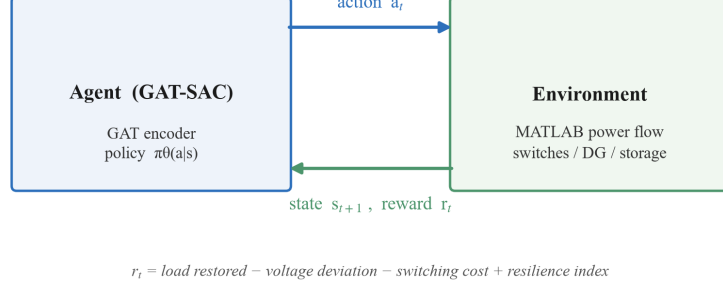


Fig. 2 Markov decision process formulation of the distribution-network self-healing decision

State. s_t consists of node injections, node voltages, branch flows, switch states, and the resilience index R , and is transformed by the graph attention encoder into a topology-aware latent representation.

Action. a_t is a discrete action combining the open/close operations of sectionalizing and tie switches with the dispatch levels of DG.

Reward. To embed resilience assessment into the decision objective, the reward is designed as

$$r_t = \lambda_1 R_t + \lambda_2 \eta_{r,t} - \lambda_3 P_{\text{loss},t} - \lambda_4 N_{\text{sw},t} \quad (7)$$

where R_t is the instantaneous resilience index, $\eta_{r,t}$ the load restoration rate, $P_{\text{loss},t}$ the network loss, $N_{\text{sw},t}$ the number of switching operations, and λ_1 – λ_4 the corresponding weights. Embedding R_t directly into the reward steers the agent to maximize resilience during training, realizing “assessment-as-objective” integrated modelling.

The state-space design determines the agent’s decision capability. Beyond continuous quantities such as node voltages, injections, and branch flows, we encode the states of all sectionalizing and tie switches as binary variables and include the instantaneous resilience index R_t so the agent perceives how well its decisions attain the resilience objective. The action space must balance discrete switching with DG dispatch; to avoid an impractically large action space, we discretize DG output into levels that, together with the switching actions, form a discrete action vector, reducing the problem to a standard discrete-action reinforcement-learning setting.

3 Topology-aware state representation via graph attention networks

3.1 Graph modelling of distribution networks

Distribution networks have a natural graph structure: buses are nodes, feeders and tie lines are edges, and DG, loads, and switches are node or edge attributes. We abstract the network as a graph $G = (V, E)$, where V and E are the node and edge sets. Each node i carries a feature vector $x_i \in \mathbb{R}^d$ containing injections, voltage magnitude, and load level; each edge (i, j) carries impedance, power flow, and switch state.

Node features comprise the voltage magnitude, active and reactive injections, load level, and DG availability of each bus; edge features comprise branch impedance, current flow, and switch state. This unified encoding of electrical quantities and topology preserves the physical structure of the network and provides rich initial features for GAT message passing. The graph structure is dynamic: a faulted branch is removed after a fault, and a new edge is added when a tie switch closes. GAT learns adaptively on such evolving graphs—a key advantage over graph convolutional networks with fixed adjacency. Fig. 3 shows the abstraction of the IEEE 33-bus system from physical topology to attributed graph, with tie switches drawn as dashed lines and DG nodes marked in colour.

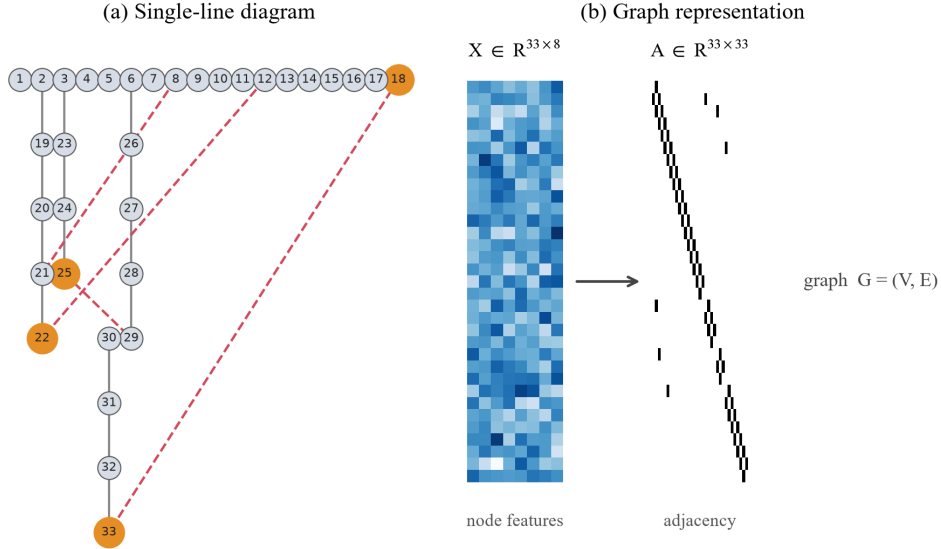


Fig. 3 Conversion of the IEEE 33-bus physical topology into an attributed graph representation

3.2 Graph attention networks

Graph attention networks (GATs) extend graph convolutional networks with an attention mechanism that assigns adaptive importance to each neighbour, capturing key nodes in the topology without relying on a fixed adjacency structure [25, 26]. For node i and a neighbour $j \in \mathcal{N}(i)$, a learnable linear transformation first computes the attention score:

$$e_{ij} = \text{LeakyReLU}(a^\top [Wh_i \parallel Wh_j]) \quad (8)$$

where h_i and h_j are the features of nodes i and j , W is a shared linear transformation, a is an attention weight vector, and $[\cdot \parallel \cdot]$ denotes concatenation. A softmax then normalizes the scores into attention coefficients α_{ij} :

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})} \quad (9)$$

Finally, the updated feature of node i is the attention-weighted aggregation of neighbour features:

$$h_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W^{(l)} h_j^{(l)} \right) \quad (10)$$

For robustness and expressiveness, GAT typically uses multi-head attention, averaging the outputs of K independent heads:

$$\tilde{h}_i^{(l+1)} = \frac{1}{K} \sum_{k=1}^K \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^k W^{(k)} h_j^{(l)} \right) \quad (11)$$

The attention mechanism lets each node assign differentiated weights to its neighbours when aggregating information. In Eq. (8), the score e_{ij} measures the importance of node j to node i ; a larger e_{ij} means node i draws more from node j . Eq. (9) normalizes the scores into probabilistic coefficients summing to one, and Eq. (10) updates node i by a nonlinear activation of the weighted neighbour features. Unlike graph convolutional networks with fixed normalized adjacency, GAT weights adapt to the input features and can therefore identify faulted regions, source nodes, and critical tie branches under specific operating conditions. Multi-head attention captures topological correlations across subspaces and, importantly, its weight matrices provide a natural source of interpretability. Fig. 4 illustrates single- and multi-head attention aggregation.

3.3 GAT state encoder

We embed this GAT as the state encoder of the reinforcement-learning agent. At each time t , the encoder organizes the voltages, injections, and loads of all buses into a node feature matrix and the branch flows and switch states into an edge feature matrix; L GAT layers then propagate messages and aggregate neighbourhood information, producing a latent state $h_i^{(L)}$ for each node. To obtain a graph-level representation, a

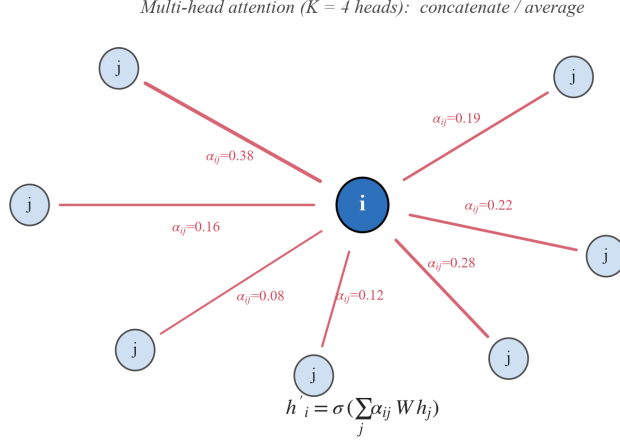


Fig. 4 Illustration of the graph attention mechanism: multi-head attention aggregation over neighbouring nodes

readout operation (mean or attention pooling) aggregates the node states into a global state vector s_g :

$$s_g = \text{Readout}\left(\{h_i^{(L)}\}_{i \in \mathcal{B}}\right) \quad (12)$$

This global vector s_g preserves topological information about critical buses and branches while compressing the representation into a low-dimensional form amenable to the policy network, providing high-quality input features for the subsequent GAT-SAC decision.

In our implementation, the encoder stacks $L = 3$ graph attention layers, each with 4 heads and a hidden dimension of 128. Residual connections and layer normalization between layers mitigate over-smoothing in deep graph networks, and the readout uses mean pooling. Because the parameters are shared across nodes, the encoder generalizes to networks of different sizes.

Compared with conventional encoders, the GAT state encoder offers two benefits. Attention lets the agent focus adaptively on faulted regions and critical transfer paths while suppressing irrelevant nodes, so decision performance remains stable under partial observability and noisy real-time measurements. And because GAT models topological interactions through message passing, the agent is intrinsically sensitive to network-structure changes and updates its awareness of connectivity and supply adequacy immediately after a switching action. Together these are the root cause of the advantage of GAT-SAC over fully connected SAC and DDPG.

4 GAT-SAC algorithm

4.1 Soft actor–critic

Soft actor–critic (SAC) is an off-policy maximum-entropy deep reinforcement learning algorithm [16, 17]. Unlike conventional reinforcement learning, which maximizes cumulative reward alone, SAC adds an entropy term to the objective to balance exploration and exploitation. The optimal policy maximizes

$$J(\pi) = \sum_t \mathbb{E}_{(s_t, a_t) \sim \rho_\pi} [r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot \mid s_t))] \quad (13)$$

where $\mathcal{H}(\pi(\cdot \mid s_t))$ is the entropy of policy π at state s_t and the temperature parameter α controls the weight of the entropy term. SAC uses an actor–critic structure: the critic learns the state–action value $Q_\theta(s, a)$ via the soft Bellman operator, and the actor updates the policy π_ϕ by minimizing KL divergence. The soft value function is updated toward

$$J_Q(\theta) = \mathbb{E} \left[\frac{1}{2} (Q_\theta(s, a) - (r + \gamma \mathbb{E}_{a'} [Q_{\bar{\theta}}(s', a') - \alpha \log \pi_\phi(a' \mid s')]))^2 \right] \quad (14)$$

where $Q_{\bar{\theta}}$ is the target network and γ the discount factor. The temperature α is also learned adaptively to regulate exploration during training [17]. Maximum entropy endows SAC with good sample efficiency and exploration, making it suitable for the large, tightly constrained action spaces of distribution-network self-healing.

The maximum-entropy objective is the defining difference between SAC and other reinforcement-learning algorithms. Conventional reinforcement learning, by maximizing reward alone, can converge prematurely to a single deterministic policy and under-explore in complex multi-modal environments. By adding the policy entropy term, SAC keeps some stochasticity in the policy, explores the action space more thoroughly during training, and avoids poor local optima. The temperature α sets the exploration–exploitation trade-off: a larger α yields a more random, more exploratory policy, and a smaller α a more deterministic, more exploitative one. Because α is adapted automatically, SAC balances the two across training stages, improving both stability and final policy quality.

4.2 Deep integration of GAT and SAC

We use the GAT as the shared backbone of the SAC actor and critic, yielding the GAT-SAC algorithm (Fig. 5). Unlike conventional SAC with fully connected backbones, GAT-SAC first encodes raw observations into a topology-aware graph representation and then feeds this representation into the policy and value heads for decision and evaluation. In the actor, the global state vector s_g from the GAT encoder is mapped by a multilayer perceptron into action-distribution parameters; discrete switching actions are sampled via Gumbel–Softmax and DG dispatch via a Gaussian policy. In the critic, s_g is concatenated with the action a and passed to the value network, which outputs Q and V values. Because the actor and critic share one GAT backbone,

they learn topological features jointly, markedly improving the policy’s adaptability to network-structure changes and its generalization.

The action space has two parts: open/close operations on 32 sectionalizing and 5 tie switches (a 37-dimensional binary action) and dispatch-level adjustments of 4 DG units. For high-dimensional discrete actions, the actor outputs a probability distribution over switching operations and dispatch levels; Gumbel–Softmax reparameterization provides differentiable sampling during training so that gradients propagate back to the GAT backbone. The critic takes both the state representation and the action vector and outputs the state–action value that drives policy updates. Because the actor and critic share the GAT backbone, they reinforce each other in backpropagation, enriching the learned topological features:

$$\pi_\phi(a | s) = f_\pi(\text{GAT}(s)), \quad Q_\theta(s, a) = f_Q(\text{GAT}(s), a) \quad (15)$$

Embedding the GAT inside SAC as a shared backbone is motivated by the nature of self-healing: it is a sequential decision problem on a graph, whose optimal policy depends on precise modelling of topological connectivity and electrical coupling. If a graph encoder were used only at the input while the policy and value networks remained fully connected, topological information would attenuate through the later layers; sharing one GAT backbone keeps the actor and critic in a common topological feature space, avoids redundant feature computation, and aligns value estimation with policy updates on the same topological prior.

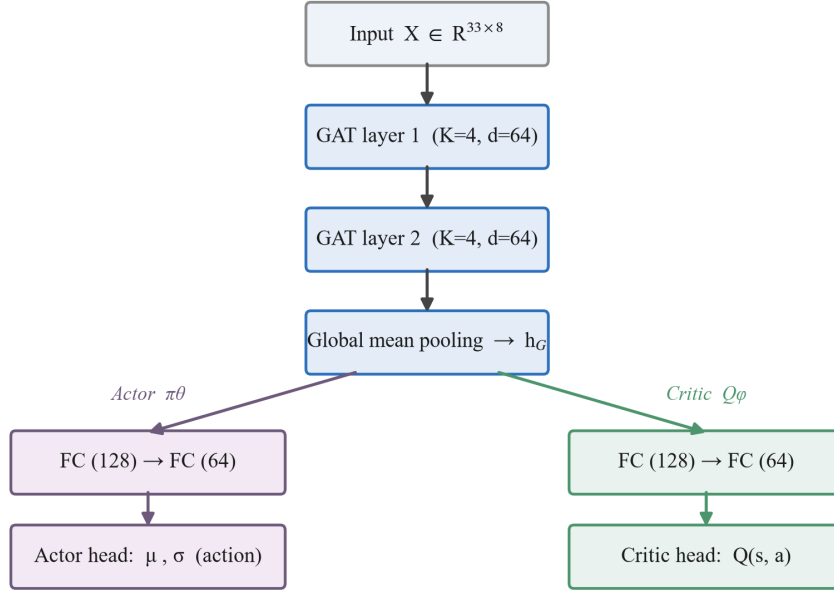


Fig. 5 Architecture of the proposed GAT-SAC algorithm with a shared graph attention backbone

4.3 Dual-track explainability

To counter the black-box nature of conventional reinforcement learning, we propose a dual-track explainability mechanism—*intrinsic attention plus post-hoc SHAP* (Fig. 6)—that explains decisions both from inside the model and by post-hoc attribution.

The first track is *intrinsic attention*. The multi-head attention coefficients α_{ij} of the GAT directly reflect the dependence of node i on its neighbour j ; visualizing the attention weights over the network topology yields an attention heatmap that reveals which buses and branches the agent focuses on—answering *where* it looks. This explanation comes from the model structure itself, needs no extra computation, and is real-time and faithful.

The second track is *post-hoc SHAP*. SHAP decomposes a model output into the marginal contributions of input features through Shapley values [32, 33]. For the action value or probability output by the policy network, the Shapley value of feature x_m is

$$\phi_m = \sum_{S \subseteq F \setminus \{m\}} \frac{|S|! (|F| - |S| - 1)!}{|F|!} [f(S \cup \{m\}) - f(S)] \quad (16)$$

where F is the full feature set, S is a subset excluding m , and $f(S)$ is the model output using only subset S . The sign and magnitude of ϕ_m capture the direction and importance of feature m , answering *why*. We combine TreeSHAP and KernelSHAP to balance attribution accuracy and computation.

In practice, *intrinsic attention* reuses the last-layer multi-head attention weights of the GAT encoder and renders them as a heatmap over the IEEE 33-bus topology, with larger weights marking buses or branches more critical to the decision; this explanation is available at every decision with negligible overhead. *Post-hoc SHAP* estimates the Shapley value of each input feature for the action value through multiple forward passes of the policy network and ranks features by absolute value. To limit cost, we approximate Shapley values with background-sample sampling and feature grouping, keeping single-decision explanation time within an acceptable range. The two tracks corroborate each other: *intrinsic attention* gives real-time, fine-grained topological attention, and SHAP gives a global, quantitative feature-importance ranking. Together they provide dispatchers with trustworthy decision evidence.

4.4 Training procedure

GAT-SAC trains offline and decides online. During training, the agent interacts with the environment under many random fault scenarios; the environment, driven by the MATLAB power-flow engine, returns voltages, losses, and load restoration in real time, and the agent updates its parameters with the reward in Eq. (7). After convergence the policy network is frozen and used for online inference, giving millisecond-level self-healing decisions. Key hyperparameters are listed in Table 2.

In each training episode, the environment generates a random fault scenario; the agent starts from the post-fault initial state and executes switching and dispatch actions until the network is radial and the load-restoration rate stops improving, or a

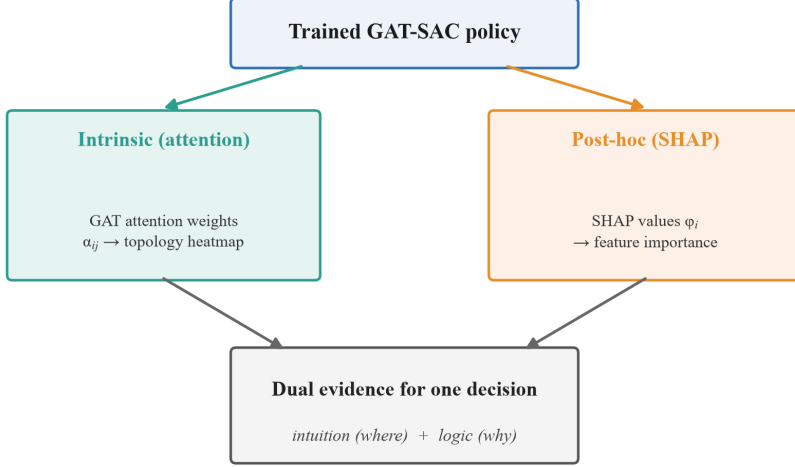


Fig. 6 Dual-track explainability mechanism: intrinsic GAT attention heatmap and post-hoc SHAP attribution

Table 2 Key hyperparameters of the GAT-SAC algorithm

Hyperparameter	Symbol	Value
GAT layers	L	3
Attention heads	K	4
Hidden dimension	d	128
Discount factor	γ	0.99
Initial temperature	α_0	0.2
Learning rate	η	3×10^{-4}
Replay-buffer capacity	D	1×10^6
Batch size	B	256

step limit is reached. Experience samples, stored as tuples (s_t, a_t, r_t, s_{t+1}) in a replay buffer, are sampled in mini-batches to update the networks. To stabilize training, the critic uses twin Q-networks and target networks whose parameters are updated by soft replacement. Training runs for 600 episodes until the cumulative reward converges.

5 Results

5.1 Simulation platform and test systems

We built a MATLAB–Python co-simulation platform (Fig. 7) [39]. MATLAB R2025b performs power-flow computation and environment simulation, solving steady-state power flow for the IEEE 33-bus and 123-bus systems with the backward/forward sweep method [40] and returning node voltages, branch flows, losses, and load restoration in

real time. The Python side builds the GAT-SAC agent with PyTorch, exchanges data through NumPy and Pandas, and visualizes results with Matplotlib. The two sides interact through CSV interfaces.

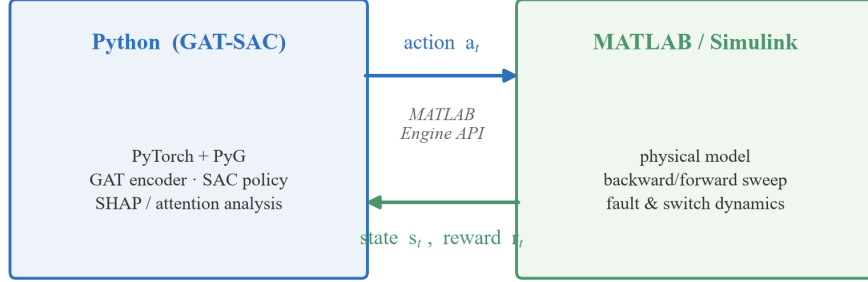


Fig. 7 MATLAB and Python co-simulation platform for distribution-network resilience assessment

The platform uses a loosely coupled architecture: MATLAB solves the physical model—fault simulation, network reconfiguration after switching, and backward/forward sweep power flow—while Python handles training and inference. Data pass through CSV files; at each step Python writes the chosen action to CSV, and MATLAB reads it, runs the power flow, and writes back the new state and reward. This design exploits the maturity of MATLAB in power-system simulation and the ecosystem of Python in deep learning while decoupling physical simulation from learning.

The test systems are the IEEE 33-bus and 123-bus benchmarks (Table 3) [41, 42]. The 33-bus system has 32 sectionalizing and 5 tie branches, with 4 DG units at buses 18, 22, 25, and 33; the 123-bus system is larger and is used to test scalability.

5.2 Scenarios and baselines

We set two fault scenarios. Scenario 1 is a single-branch fault opening branch 12–13, de-energizing part of the downstream load. Scenario 2 is a multi-branch fault opening branches 2–19 and 6–26 simultaneously, with a larger de-energized region and greater restoration difficulty. Self-healing closes tie switches to reconfigure the topology and dispatches DG for load transfer.

We compare against three baselines: (1) standard SAC with a fully connected policy backbone; (2) DDPG, a deterministic policy-gradient algorithm; and (3) mixed-integer

Table 3 Main parameters of the test systems

Parameter	IEEE 33	IEEE 123
Number of buses	33	123
Branches (sectional/tie)	37 (32/5)	117
Rated voltage (kV)	12.66	4.16
Total load (MW)	3.715	3.49
DG units	4	5
DG capacity (kW)	1050	1250

linear programming (MILP), the benchmark for decision quality. The configurations are summarized in Table 4.

Table 4 Baseline algorithms

Algorithm	Policy network	Description
GAT-SAC	GAT + MLP	Proposed; topology-aware, maximum entropy
SAC	Fully connected MLP	Baseline without topology awareness
DDPG	Fully connected MLP	Deterministic-policy baseline
MILP	—	Mathematical-programming optimum

5.3 Resilience assessment

Fig. 8 and Fig. 9 show the resilience performance curves under the two scenarios, and Table 5 summarizes the key metrics. MILP attains the theoretically highest restoration by global search, with the best minimum-performance and recovery levels. GAT-SAC is close to MILP: in Scenario 1 its minimum performance level is 0.65 and recovery level 0.98, nearly matching MILP (0.65/0.99), with a shorter normalized recovery time (0.50 vs. 0.52). SAC and DDPG, lacking topology awareness, degrade more deeply and recover more slowly, giving clearly lower resilience.

Fig. 8 shows that all node voltages lie within $[0.95, 1.05]$ in normal operation; after the fault, the voltages of downstream nodes drop sharply or collapse, and after self-healing they return to the normal band. The voltage profile restored by GAT-SAC closely matches MILP, whereas SAC and DDPG leave some nodes undervoltage, indicating suboptimal transfer paths with detoured flows or insufficient transfer capacity. This physically corroborates the effectiveness of GAT topology-aware representation.

Under the multi-branch fault of Scenario 2, all methods degrade, but GAT-SAC still approaches MILP (recovery level 0.97 vs. 0.98), substantially outperforming SAC (0.85) and DDPG (0.78). The GAT topology-aware representation enables the agent to find near-optimal restoration policies under more complex fault combinations, demonstrating good generalization.

Auxiliary metrics corroborate this. GAT-SAC tracks MILP in load-restoration rate and mean restoration time in both scenarios, with a slight edge in recovery speed;

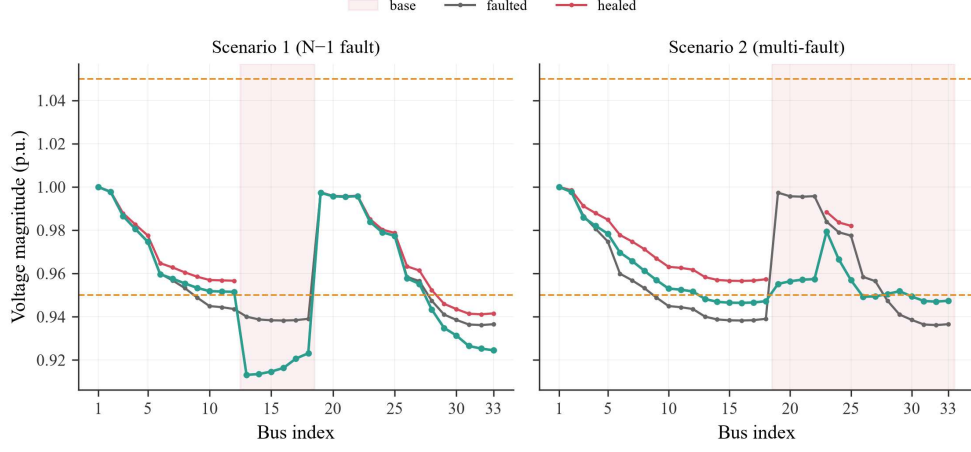


Fig. 8 Voltage profiles of the IEEE 33-bus system before and after fault self-healing

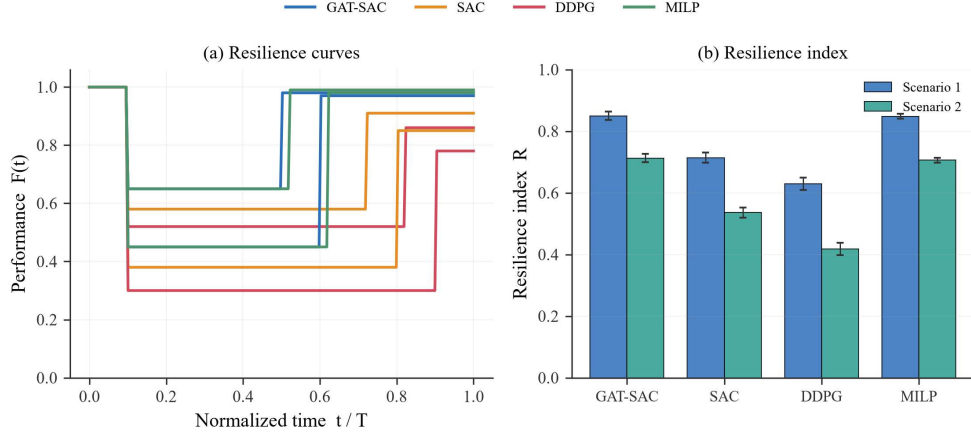


Fig. 9 Resilience performance curves under the two fault scenarios for the different methods

SAC and DDPG deteriorate markedly in both completeness and speed under multi-branch faults. The resilience index R in Table 5 integrates these differences into a single comparison.

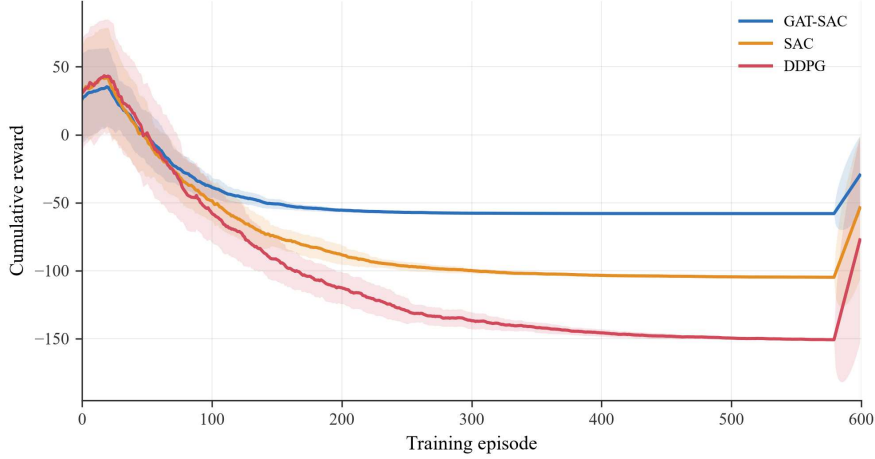
5.4 Decision quality and real-time performance

Fig. 10 shows the cumulative-reward convergence curves. GAT-SAC, benefiting from topology awareness and maximum-entropy exploration, converges fastest and to the highest reward; SAC is next; DDPG, with deterministic exploration, converges slowly and to a lower reward.

The curves reveal the mechanism differences. GAT-SAC reaches a high reward within about 150 episodes and then converges smoothly; SAC converges more slowly to

Table 5 Comparison of resilience indicators

Method	Scenario 1			Scenario 2		
	$F_{\min}$	F_{rec}	t_r	$F_{\min}$	F_{rec}	t_r
GAT-SAC	0.65	0.98	0.50	0.45	0.97	0.60
SAC	0.58	0.91	0.72	0.38	0.85	0.80
DDPG	0.52	0.86	0.82	0.30	0.78	0.90
MILP	0.65	0.99	0.52	0.45	0.98	0.62

**Fig. 10** Training convergence curves of cumulative reward for GAT-SAC, SAC, and DDPG

a clearly lower reward; and DDPG, limited by deterministic exploration in a discrete action space, converges slowly and fluctuates. The 2.8% optimality gap of GAT-SAC arises mainly from action-space discretization and could shrink with finer DG dispatch levels.

Table 6 compares online decision time and the optimality gap relative to MILP. GAT-SAC decides in 2.3 ms, roughly three orders of magnitude faster than MILP (~ 1.2 s), with only a 2.8% optimality gap. On the IEEE 123-bus system, MILP exceeds 10 s while GAT-SAC remains millisecond-level, underscoring its real-time advantage.

Table 6 Online decision time and optimality gap relative to MILP

Method	Online decision time (ms)	Optimality gap vs. MILP (%)
GAT-SAC	2.3	2.8
SAC	2.1	5.6
DDPG	1.9	8.4
MILP	1.2×10^3	0 (reference)

5.5 Explainability analysis

Fig. 11 shows the last-layer attention heatmap of GAT-SAC in the fault scenario. The agent focuses on the faulted branch 12–13 and adjacent buses 12 and 13, as well as tie switch 18–33 and DG node 25 that provide transfer paths—consistent with dispatcher intuition.

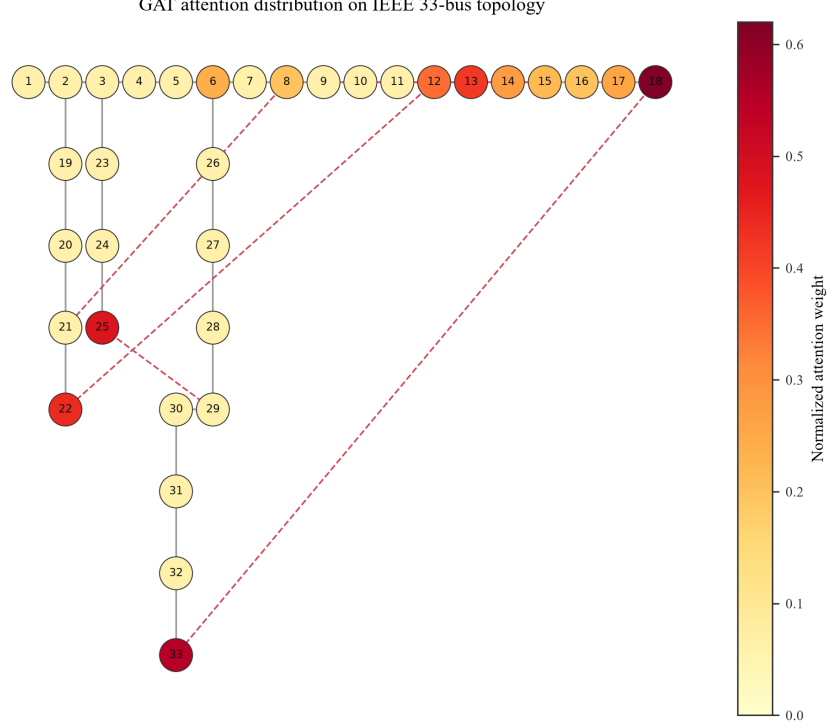


Fig. 11 GAT multi-head attention heatmap highlighting critical buses and branches for self-healing

Fig. 12 shows the SHAP feature-importance ranking. Bus-18 voltage, branch 7–8 flow, bus-33 voltage, tie-switch 18–33 state, and DG-25 output are the top five features; bus-18 voltage has the highest SHAP value (0.32), indicating that the supply adequacy of its feeder dominates the restoration decision. The SHAP attribution corroborates the attention heatmap.

Attention and SHAP rankings reveal a consistent pattern: the agent concentrates on the faulted region and its available transfer paths—buses near faulted branch 12–13, tie switch 18–33, and DG 25. This matches the dispatcher’s heuristic of prioritizing critical loads and exploiting nearby ties and DG, showing that the agent not only decides correctly but does so for comprehensible, verifiable reasons. Such explainability is essential in safety-critical, accountability-heavy power systems and is a prerequisite for engineering deployment.

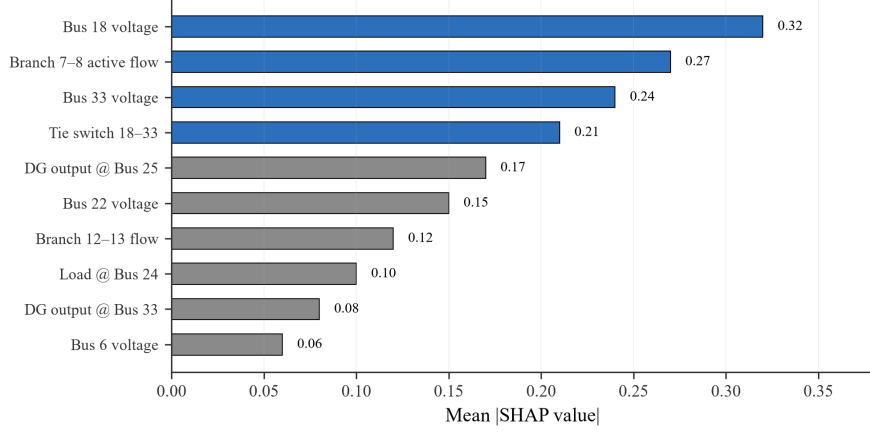


Fig. 12 SHAP feature-importance ranking for the self-healing decision of GAT-SAC

Fig. 13 shows the end-to-end explainable decision chain from fault occurrence to resilience recovery—fault localization, attention focus, action selection, SHAP attribution, and resilience recovery—providing dispatchers a complete decision trail that strengthens trust in the intelligent decision system.

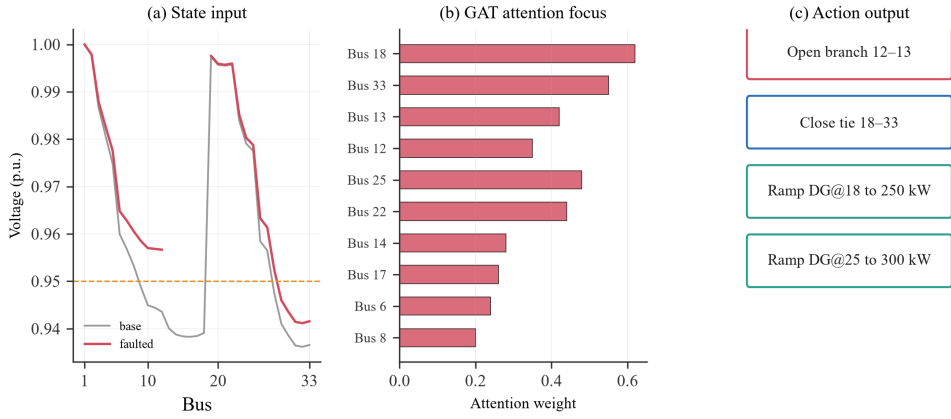


Fig. 13 End-to-end explainable decision chain from fault occurrence to resilience recovery

6 Conclusion

We presented an explainable graph reinforcement learning method for resilience assessment and self-healing decision-making in distribution networks. By modelling the network as a graph, using a graph attention network as the shared backbone of a soft actor-critic agent, and combining a dual-track explainability mechanism with

a resilience-embedded reward, we integrate resilience assessment and self-healing decision-making. Four findings emerge.

First, GAT-SAC attains decision quality within 3% of the MILP optimum and clearly outperforms SAC and DDPG, confirming that topology-aware representation improves decision quality. Second, GAT-SAC decides in milliseconds, about three orders of magnitude faster than mathematical programming, and scales to the IEEE 123-bus system. Third, the dual-track explainability mechanism outputs attention heatmaps and SHAP rankings consistent with dispatcher intuition, giving dual evidence of *where* and *why*, and increasing trust in the decision system. Fourth, embedding the resilience index into the reward realizes assessment-as-objective modelling, avoiding the decoupling of assessment from decision.

The method has limitations. The action space is discretized, leaving room for joint optimization of continuous DG output and switching; training data come from predefined fault scenarios, so generalization to unseen extreme scenarios needs validation; and global SHAP attribution does not yet capture the real-time causal chain of a single decision. Future work will introduce continuous-discrete hybrid action spaces and hierarchical reinforcement learning for higher decision precision, combine domain randomization and meta-learning to generalize to extreme scenarios, and explore causal inference and counterfactual explanation for deeper interpretability.

Overall, this work provides a technical path that balances decision quality, real-time performance, and explainability for distribution-network fault self-healing. The framework extends beyond resilience assessment and self-healing to voltage control, reactive-power optimization, and state estimation. As DG penetration and distribution automation grow, explainable intelligent decision-making will play an increasing role in modern distribution systems.

Declarations

- Competing interests: The authors declare no competing interests.
- Funding: No funding was received for conducting this study.
- Authors' contributions: Yehaoxiang Liu: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization, Project administration. Chizhen Wang: Software, Methodology, Data curation, Validation, Writing – review & editing.
- Acknowledgements: Not applicable.
- Ethics approval and consent to participate: Not applicable.
- Consent for publication: Not applicable.
- Availability of data and materials: The datasets generated during the current study are available from the corresponding author on reasonable request.

References

- [1] Yaghoubi, E. *et al.* Computational methods for joint DG-switch optimization in distribution networks: A systematic review and bibliometric analysis (2020–2025). *Archives of Computational Methods in Engineering* (2026). URL <https://doi.org/10.1007/s11831-026-10513-x>.

- [2] Rouholamini, M., Wang, C., Magableh, S. & Wang, X. Resiliency of electric power distribution networks: a review. *Journal of Infrastructure Preservation and Resilience* **6**, 39 (2025). URL <https://doi.org/10.1186/s43065-025-00154-y>.
- [3] Shi, W., Bai, C. & Wang, Z. Power distribution system blackstart restoration using renewable energy. *Nature Communications* **17**, 8498 (2026). URL <https://doi.org/10.1038/s41467-026-76692-0>.
- [4] Bai, M. *et al.* Research on distribution network fault simulation based on cyber physics system. In *2019 IEEE 3rd International Electrical and Energy Conference (CIEEC)*, 1112–1116 (2019). URL <https://doi.org/10.1109/CIEEC47146.2019.CIEEC-2019412>.
- [5] Zhang, W. *et al.* Fault classification method for distribution network optical cables based on IGWO-BP. In *Proceedings of the 2026 5th International Conference on Big Data, Information and Computer Network*, 660–664 (Association for Computing Machinery, 2026). URL <https://doi.org/10.1145/3801228.3801330>.
- [6] Qiao, J. Neural network-based expert system for identification of safety hazards in distribution network equipment. In *Proceedings of the 2025 International Conference on Simulation, Modeling and Big Data*, 201–206 (Association for Computing Machinery, 2025). URL <https://doi.org/10.1145/3768740.3768771>.
- [7] Zhang, C., Zou, Y. & Gu, X. An imitation based reinforcement learning method for power network reconfiguration under random line fault scenarios. In *2026 11th Asia Conference on Power and Electrical Engineering (ACPEE)*, 1043–1047 (2026). URL <https://doi.org/10.1109/ACPEE69242.2026.11524421>.
- [8] Wu, P., Wang, K., Chen, C. & Bie, Z. Resilience assessment methods for enabling sustainable power systems with high renewable energy penetration: Toward a more sustainable energy future. *IEEE Energy Sustainability Magazine* **1**, 124–138 (2025). URL <https://doi.org/10.1109/ESM.2025.3583845>.
- [9] Gkoktsis, G., Schellien, T., Peters, L. & Eckel, M. *The cyber-resilience index: Quantification and measurement of resilience for virtual power plants, a systems engineering approach*. In *Proceedings of the 41st ACM/SIGAPP Symposium on Applied Computing*, 1463–1472 (Association for Computing Machinery, 2026). URL <https://doi.org/10.1145/3748522.3779924>.
- [10] Ushashree, P. & Kumar, K. S. Power system reconfiguration in distribution system for loss minimization using optimization techniques: A review. *Wireless Personal Communications* **128**, 1907–1940 (2023). URL <https://doi.org/10.1007/s11277-022-10026-3>.
- [11] Mutluri, R. B. & Saxena, D. A comprehensive overview and future perspectives of networked microgrids for emerging power systems. *Smart Grids and Sustainable Energy* **9**, 45 (2024). URL <https://doi.org/10.1007/s40866-024-00218-0>.

- [12] Chen, X., Han, X. & Li, T. *Research on distribution network fault restoration considering mobile resource optimization scheduling*. In *2025 International Conference on Applied Electrical Engineering and Technology (AEET)*, 252–257 (2025). URL <https://doi.org/10.1109/AEET66561.2025.11307178>.
- [13] Zhang, H., Zhao, Q. & Li, S. *Construction and analysis of low-transparency RSBFs based on the idea of PSO algorithm*. In *Proceedings of the 2024 7th International Conference on Artificial Intelligence and Pattern Recognition*, 964–971 (Association for Computing Machinery, 2025). URL <https://doi.org/10.1145/3703935.3704040>.
- [14] Kolb, L., Panzer, M. & Gronau, N. Assessing generalizability in deep reinforcement learning based assembly: A comprehensive review. *Journal of Intelligent Manufacturing* **37**, 237–255 (2026). URL <https://doi.org/10.1007/s10845-024-02546-5>.
- [15] Park, J., Jung, H., Kim, J. W. & Lee, J. M. Reinforcement learning for process control: Review and benchmark problems. *International Journal of Control, Automation and Systems* **23**, 1–40 (2025). URL <https://doi.org/10.1007/s12555-024-0990-1>.
- [16] Wong, A., de Nobel, J., Bäck, T., Plaat, A. & Kononova, A. Solving deep reinforcement learning tasks with evolution strategies and linear policy networks. *ACM Transactions on Evolutionary Learning and Optimization* (2026). URL <https://doi.org/10.1145/3831362>. Just Accepted.
- [17] Wang, H., Du, W., Xu, Y. & Cui, L. *SAC-B: Soft actor-critic with bias for suppressing Q-Value overestimation in off-policy reinforcement learning*. In *Proceedings of the 2025 7th International Conference on Distributed Artificial Intelligence*, 58–66 (Association for Computing Machinery, 2025). URL <https://doi.org/10.1145/3772429.3772436>.
- [18] Arulkumaran, K., Deisenroth, M. P., Brundage, M. & Bharath, A. A. A brief survey of deep reinforcement learning. *IEEE Signal Processing Magazine* **34**, 26–38 (2017). URL <https://doi.org/10.1109/MSP.2017.2743240>.
- [19] Hu, D. *et al.* Multi-agent deep reinforcement learning for voltage control with coordinated active and reactive power optimization. *IEEE Transactions on Smart Grid* **13**, 4873–4886 (2022). URL <https://doi.org/10.1109/TSG.2022.3185975>.
- [20] El-Fergany, A. A. Reviews, challenges, and insights on computational methods for network reconfigurations in smart electricity distribution networks. *Archives of Computational Methods in Engineering* **31**, 1233–1253 (2024). URL <https://doi.org/10.1007/s11831-023-10007-0>.
- [21] Doğan, Y. & Ünlü, R. Machine learning for V2X-enabled microgrids: A bibliometric and thematic review of intelligent energy management applications.

- Arabian Journal for Science and Engineering* **51**, 10429–10465 (2026). URL <https://doi.org/10.1007/s13369-026-11201-5>.
- [22] Guo, Z., Lai, J. & Kou, X. Intelligent distribution network active protection and fault recovery mechanism based on DRL and graph optimization. *Electric Power Systems Research* **254**, 112623 (2026). URL <https://doi.org/10.1016/j.epsr.2025.112623>.
 - [23] Liao, J. *et al.* A review on multi-types analysis for graph neural networks-based recommendation systems. *International Journal of Computational Intelligence Systems* **19**, 94 (2026). URL <https://doi.org/10.1007/s44196-026-01198-y>.
 - [24] Zolman, N., Lagemann, C., Fasel, U., Kutz, J. N. & Brunton, S. L. SINDy-RL for interpretable and efficient model-based reinforcement learning. *Nature Communications* **16**, 10714 (2025). URL <https://doi.org/10.1038/s41467-025-65738-4>.
 - [25] Song, W. & Muntaha, M. Collaborative task allocation with reinforcement learning and heterogeneous graph attention. *IEEE Transactions on Network Science and Engineering* **13**, 8313–8330 (2026). URL <https://doi.org/10.1109/TNSE.2026.3667470>.
 - [26] Chen, J., Ma, T. & Xiao, C. FastGCN: Fast learning with graph convolutional networks via importance sampling (2018). URL <https://arxiv.org/abs/1801.10247>. [arXiv:1801.10247](https://arxiv.org/abs/1801.10247).
 - [27] Haug, T. & Goebel, C. *DSML: A comprehensive framework for evaluating ML methods for distribution system state estimation*. In *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*, 122–135 (Association for Computing Machinery, 2025). URL <https://doi.org/10.1145/3679240.3734629>.
 - [28] Tam, P., Ros, S., Song, I., Kang, S. & Kim, S. A survey of intelligent end-to-end networking solutions: Integrating graph neural networks and deep reinforcement learning approaches. *Electronics* **13**, 994 (2024). URL <https://doi.org/10.3390/electronics13050994>.
 - [29] Eltwab, A. A. A. & Abdelsalam, M. M. Advanced supervisory and distributed control frameworks in modern industrial and energy applications. *Journal of Electrical Systems and Information Technology* **13**, 77 (2026). URL <https://doi.org/10.1186/s43067-026-00381-7>.
 - [30] Ghaemipour, A., Mashhadi, H. R. & Mostafavi, S. H. Accelerating the learning process of deep reinforcement learning algorithms in distribution network reconfiguration using an innovative replay method. *Scientific Reports* **16**, 12660 (2026). URL <https://doi.org/10.1038/s41598-026-40508-4>.

- [31] Xiang, Y., Lu, Y. & Liu, J. Deep reinforcement learning based topology-aware voltage regulation of distribution networks with distributed energy storage. *Applied Energy* **332**, 120510 (2023). URL <https://doi.org/10.1016/j.apenergy.2022.120510>.
- [32] Chen, H., Covert, I. C., Lundberg, S. M. & Lee, S.-I. Algorithms to estimate Shapley value feature attributions. *Nature Machine Intelligence* **5**, 590–601 (2023). URL <https://doi.org/10.1038/s42256-023-00657-x>.
- [33] Parisineni, S. R. A. & Pal, M. Enhancing trust and interpretability of complex machine learning models using local interpretable model agnostic shap explanations. *International Journal of Data Science and Analytics* **18**, 457–466 (2024). URL <https://doi.org/10.1007/s41060-023-00458-w>.
- [34] Raza, R., Wang, G., Wong, K. W., Laga, H. & Fisichella, M. *ITL-LIME: Instance-based transfer learning for enhancing local explanations in low-resource data settings*. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, 2482–2492 (Association for Computing Machinery, 2025). URL <https://doi.org/10.1145/3746252.3761183>.
- [35] Şahin, E., Arslan, N. N. & Özdemir, D. Unlocking the black box: An in-depth review on interpretability, explainability, and reliability in deep learning. *Neural Computing and Applications* **37**, 859–965 (2025). URL <https://doi.org/10.1007/s00521-024-10437-2>.
- [36] Khan, F. M. *et al.* XAI-driven data mining for self-defending IoT systems: Enhancing cybersecurity transparency in the age of smart cities. *Cognitive Computation* **18**, 16 (2026). URL <https://doi.org/10.1007/s12559-026-10559-w>.
- [37] Islam, M. R. *et al.* *Explainable multi-stage self-supervised learning framework for intelligent fault diagnosis*. In *Proceedings of the 2026 10th International Conference on Artificial Intelligence, Automation and Control Technologies*, 25–35 (Association for Computing Machinery, 2026). URL <https://doi.org/10.1145/3795496.3795710>.
- [38] Jain, K., Kumar, S., Sapra, V. & Agarwal, A. Explainable artificial intelligence (XAI) in IoT and WSN security: A systematic review. *Archives of Computational Methods in Engineering* (2026). URL <https://doi.org/10.1007/s11831-026-10623-6>.
- [39] Paulus, G. N. T. & Rix, A. J. *Power system simulation software for modelling modern grids: A review*. In *Proceedings of the 2024 Southern African Sustainable Energy Conference*, 677–689 (Springer Nature Switzerland, 2026). URL https://doi.org/10.1007/978-3-032-12579-8_55.
- [40] Meteab, F., Tousi, B. & Omran, M. Efficient self-healing framework for smart distribution networks. *Scientific Reports* **15**, 31124 (2025). URL <https://doi.org/>

[10.1038/s41598-025-16929-y](https://doi.org/10.1038/s41598-025-16929-y).

- [41] Dolatabadi, S. H., Ghorbanian, M., Siano, P. & Hatziargyriou, N. D. An enhanced IEEE 33 bus benchmark test system for distribution system studies. *IEEE Transactions on Power Systems* **36**, 2565–2572 (2020). URL <https://doi.org/10.1109/TPWRS.2020.3038030>.
- [42] Schneider, K. P. *et al.* Analytic considerations and design basis for the IEEE distribution test feeders. *IEEE Transactions on Power Systems* **33**, 3181–3188 (2017). URL <https://doi.org/10.1109/TPWRS.2017.2760011>.