

Minimax Training Closes the Reconnaissance Gap in Learned Network-Robustness Design

Md Hasibuzzaman

hsbuzzaman@gmail.com

Asia University

Research Article

Keywords: network robustness, reinforcement learning, adversarial design, critical infrastructure, graph neural networks, complex networks, power grid

Posted Date: August 3rd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10414673/v1>

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

[Read Full License](#)

Additional Declarations: No competing interests reported.

Minimax Training Closes the Reconnaissance Gap in Learned Network-Robustness Design

Md Hasibuzzaman^{1*}

^{1*}Department of Computer Science and Information Engineering, Asia University, No. 500, Liufeng Road, Wufeng District, 41354, Taichung, Taiwan.

Corresponding author(s). E-mail(s): hsbuzzaman@gmail.com;

Abstract

Reinforcement learning (RL) agents can discover network topologies substantially more robust to targeted attack than classical heuristics produce, and a 2026 study reported such agents spontaneously discover “multi-backbone” structures. Every method in this line of work, learned or heuristic, is trained and evaluated against a single, fixed attack strategy chosen in advance, leaving open whether a design that excels against one attack survives an adversary who reconnoiters the topology and adapts. We introduce a minimax training objective that optimizes a network-redesign agent against the worst of a small attack portfolio, recomputed at every training step, instead of one fixed attack. Across three topology families (Barabási–Albert, Erdős–Rényi, Watts–Strogatz) and eight seeds per family (168 runs), we compare this co-adaptive paradigm against the standard single-attack paradigm, instantiated with a graph-neural-network policy trained via actor-critic RL and with simulated annealing. Minimax training reduces an “Adaptive Degradation Gap” – robustness lost when an attacker switches from the trained-against strategy to its most damaging alternative – by 74–81% (Wilcoxon signed-rank, $n = 24$, Holm-corrected $p < 0.001$, rank-biserial $r \approx 0.94\text{--}1.0$), while achieving higher worst-case robustness. Optimizer choice makes no significant difference once the objective is fixed. The effect replicates directionally on real US power-grid subgraphs, generalizes to three unseen attack types, is driven specifically by including betweenness attack in training rather than by portfolio size, and persists at double the network scale. The contribution is conceptual rather than architectural: the training objective, not the optimizer, determines whether learned network-robustness design is brittle or robust to a realistic, reconnaissance-capable adversary.

Keywords: network robustness, reinforcement learning, adversarial design, critical infrastructure, graph neural networks, complex networks, power grid

1 Introduction

Modern societies depend on networked infrastructure [1–3] – power grids, communication backbones, transportation systems – whose function collapses disproportionately when a small number of critical nodes are removed [4, 5], sometimes triggering cascading failure well beyond the initial loss [6]. Because the cost of adding capacity is high, a large literature asks a narrower question: given a fixed budget of structural changes, which edges should be added or rewired to make an existing network as robust as possible to attack? Schneider et al. formalized this with the robustness index R , the area under the giant-component-size curve as nodes are removed in order of decreasing importance, and showed that simulated annealing (SA) [7] over degree-preserving edge swaps can improve R substantially at low cost [8].

The last five years have replaced hand-designed heuristics with learned ones. RNet-DQN cast graph construction as a Markov decision process solved with deep Q-learning [9] and graph neural networks (GNNs) [10]. ResiNet reformulated resilience optimization as sequential edge rewiring learned end-to-end with policy gradients [11]. Most recently, Zhu et al. trained a GNN-based policy with reinforcement learning to redesign networks and reported the spontaneous emergence of redundant “multi-backbone” structures that classical heuristics rarely discover [12].

A structural assumption runs through every one of these methods, learned or classical: the redesign agent is trained and evaluated against *one fixed attack strategy*, typically targeted removal of the highest-degree or highest-betweenness nodes, computed once and applied without further adaptation. Zhu et al. explicitly identify this as a boundary of their own work, distinguishing the *design stage* (before any attack occurs, the setting all of the above methods address) from an *operational stage* in which a handful of game-theoretic studies model an attacker and defender who interact after the topology is fixed. They note that closing this gap – letting the attacker adapt *during* the design process itself – is future work.

This matters because real adversaries reconnoiter before they strike. An attacker targeting critical infrastructure can observe the deployed topology and choose whichever attack vector it is most vulnerable to; a defense over-fit to degree-based attack (a common training target because it is cheap to compute) can leave a network quietly exposed to betweenness-based attack on the same graph. If this is a general phenomenon rather than an artefact of any one topology or method, then a substantial fraction of the network-robustness-design literature has been optimizing a proxy that a realistic adversary would not respect.

We test this directly. Rather than training a redesign agent against one fixed attack, we train it against the worst-of-a-portfolio of attacks, recomputed at every rewiring step – a minimax objective inspired by, but considerably lighter-weight than, the two-player self-play framework of robust adversarial reinforcement learning [13]. We compare this against the standard single-attack paradigm using two independent optimizers (a GNN actor-critic policy and simulated annealing) so that any observed effect can be attributed to the *training objective* rather than to a particular solver. Our central empirical question is simple and falsifiable: does training against a single fixed attack produce designs that are quietly brittle to a reconnaissance-capable adversary, and does minimax training fix this without sacrificing raw robustness?

2 Related work

2.1 Robustness metrics and classical design

The robustness index $R(G) = \frac{1}{N} \sum_{Q=0}^{N-1} s(Q)$, where $s(Q)$ is the fraction of nodes in the largest connected component after removing the Q highest-priority nodes, was introduced by Schneider et al. together with hill-climbing and simulated-annealing procedures for improving it under degree-preserving edge swaps [8], building on percolation-based robustness measures [14] and immunization strategies that target structurally important nodes [15]. Attack strategies ranked by degree (HDA) or betweenness (HBA), and whether the ranking is fixed at the outset or recalculated after each removal, were characterized by Albert et al. and Holme et al. [4, 5]. Alenazi et al. studied cost-constrained backbone design that balances algebraic connectivity against centrality-driven vulnerability, a classical (non-learned) precursor to the degree-homogenization heuristic we use as a baseline [16].

2.2 Learned network redesign

RNet-DQN [10] and ResiNet [11] independently established that GNN-based agents trained with reinforcement learning can discover edge modifications that outperform degree- and betweenness-based heuristics, evaluating primarily against the attack objective used for training. Zhu et al. scaled this approach and, notably, performed cross-attack evaluation showing that policies trained against one attack transfer imperfectly to others – a finding consistent with our results, though their training loop itself still targets a fixed attack per run [12]. All three works optimize an *inductive* policy intended to generalize across many graph instances; because our contribution is the training objective rather than generalization, we instead optimize instance-specific policies (Section 3), a simplification discussed as a limitation in Section 6.

2.3 Adversarial and game-theoretic framings

Robust adversarial reinforcement learning (RARL) trains a protagonist policy against a learned adversary that perturbs the environment, improving robustness to model mismatch in continuous-control tasks [13]. We adopt its central idea, that training against an adapting opponent yields more robust policies than training against a fixed one, but apply a considerably lighter-weight instantiation: rather than a second trained neural adversary, our attacker selects, at every step, the worst of a small fixed portfolio of well-understood attack heuristics. This keeps training tractable while still operationalizing genuine within-training adaptivity, and we discuss the trade-off against a fully learned adversary in Section 6.

3 Methods

Figure 1 summarizes the full pipeline: a damaged network is rewired under either a fixed single-attack objective or the minimax objective introduced here, using either an RL policy or simulated annealing as the optimizer, and every resulting design is

evaluated under a common multi-attack protocol. The remainder of this section details each stage.

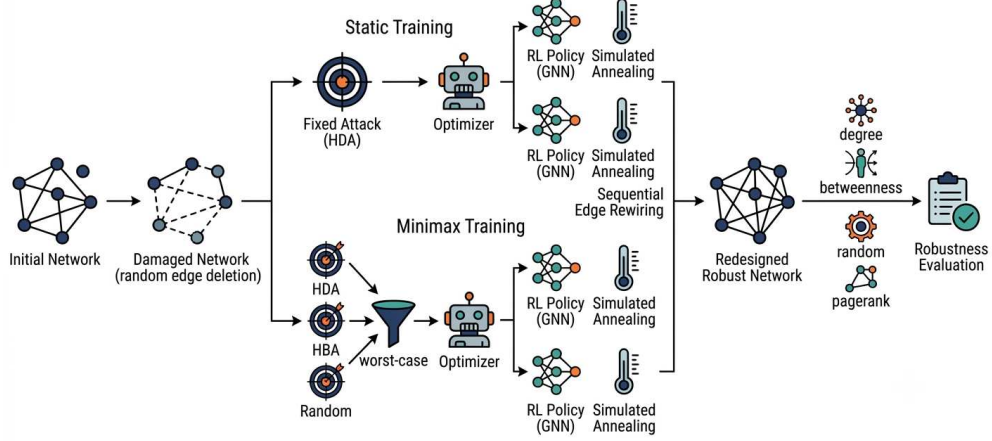


Fig. 1 Overview of the redesign and evaluation pipeline. An initial network is damaged by random edge deletion, then rewired under a static (single fixed-attack) or minimax (worst-of-portfolio) training objective, each solved by both an RL policy and simulated annealing. The four resulting designs are evaluated under a common attack portfolio (degree, betweenness, random, PageRank).

3.1 Robustness metric and attack strategies

We use the Schneider robustness index $R(G)$ under three attack strategies, each *recalculated* after every node removal (i.e. the ranking is recomputed on the residual graph, not fixed at the outset): high-degree attack (HDA), high-betweenness attack (HBA) [17], and random removal (averaged over repeated trials). Given a portfolio of strategies $\mathcal{A} = \{\text{HDA}, \text{HBA}, \text{Random}\}$, we define the *worst-case robustness*

$$R_{\text{worst}}(G) = \min_{a \in \mathcal{A}} R_a(G), \quad (1)$$

representing a reconnaissance-capable attacker that selects whichever known strategy is most damaging to the specific topology it observes. We additionally report the *Adaptive Degradation Gap*,

$$\Delta(G) = R_{\text{HDA}}(G) - R_{\text{worst}}(G), \quad (2)$$

which is zero when a design is equally robust to every attack in the portfolio and grows as the design becomes brittle to whichever attack it was not optimized for. $\mathcal{A}$

is also the *training* portfolio used throughout Section 4. To test whether minimax training against this small portfolio generalizes to attacks it never saw, Section 4.4 additionally evaluates every design under an *expanded, evaluation-only* portfolio $\mathcal{A}^+ = \mathcal{A} \cup \{\text{PageRank}[18], \text{Eigenvector}[19], k\text{-shell}[20]\}$, yielding R_{worst}^+ and Δ^+ defined analogously over $\mathcal{A}^+$. A fully learned or collective-influence attacker [21] is a natural further extension, discussed in Section 6.

3.2 Rewiring mechanics

Following the decomposition used by Zhu et al. [12], we separate each rewiring episode into a random deletion stage and a strategic addition stage. Starting from an initial graph G_0 , a fraction α of edges is deleted uniformly at random, producing a damaged instance H_0 ; a budget of B edges is then added back one at a time, either by a baseline heuristic, simulated annealing, or a learned policy, holding total edge count approximately fixed across methods for a fair comparison. This isolates the *addition* decision, the object of interest, from the deletion process, which is identical and non-strategic across all methods.

3.3 Defender policy

The addition policy is a two-layer graph convolutional network [22] over four structural node features (degree, clustering coefficient, average neighbor degree, k -core number, each min-max normalized), producing per-node embeddings z_v . At each step, every legal non-edge (u, v) is scored by a small MLP over $[z_u \odot z_v, |z_u - z_v|]$, and an edge is sampled from the resulting softmax distribution (actor). A separate MLP maps the mean-pooled graph embedding to a scalar value estimate (critic). Training uses single-trajectory advantage actor-critic (A2C) [23] with entropy regularization and discounted returns, rather than the full clipped-objective PPO [24] used in [10, 12]; this is a deliberate scope choice for implementation reliability within our compute budget, discussed further in Section 6. Because our contribution targets the training *objective* rather than generalization across a graph distribution, each policy is optimized *on one fixed damaged instance* H_0 over many rollouts, analogous to how the SA baseline also searches a single instance; the best-performing rollout observed during optimization is reported (a standard reporting convention in neural combinatorial optimization).

3.4 Training regimes

We compare two training objectives, each usable with either optimizer:

- **Static:** reward is the step-wise improvement in $R_{\text{HDA}}(G)$ alone, matching the single-fixed-attack paradigm of prior work.
- **Minimax:** reward is the step-wise improvement in $R_{\text{worst}}(G)$ (Eq. 1), i.e. the attacker is allowed to reselect its most damaging strategy at every step as the topology changes.

Applying both objectives to two independent optimizers yields four design methods: RL-STATIC, RL-MINIMAX, SA-STATIC, SA-MINIMAX. We additionally report a degree-homogenizing greedy baseline (DCBO, connecting the lowest-degree available

node pairs at each step, in the spirit of [16]), a best-of-30-restarts RANDOM addition baseline, and the undefended damaged graph (NoDESIGN) as a reference floor.

3.5 Experimental setup

Initial graphs G_0 ($N = 50$ nodes, mean degree ≈ 4) are generated from three canonical families: Barabási–Albert (BA) [25], Erdős–Rényi (ER) [26], and Watts–Strogatz (WS) [27], spanning scale-free, homogeneous-random, and small-world regimes. For each family we draw 8 independent seeds (used for both graph generation and the random deletion draw), giving 24 independent problem instances. Deletion fraction $\alpha = 0.3$; addition budget $B = 15$ edges. RL policies are optimized for 200 rollouts per instance; SA runs for 300 (static objective) or 250 (minimax objective) proposal steps, matched roughly for wall-clock cost given the higher per-evaluation cost of the minimax objective. All seven methods are evaluated identically post hoc under R_{HDA} , R_{HBA} , R_{random} , R_{worst} , network efficiency, algebraic connectivity, and the Adaptive Degradation Gap, regardless of what each method was optimized for, so that comparisons are not biased toward any one method’s own training objective. Paired comparisons across the 24 matched instances use the Wilcoxon signed-rank test, the matched-pairs rank-biserial correlation as an effect size, a 10,000-sample bootstrap 95% confidence interval on the mean paired difference, and Holm–Bonferroni correction across the family of eight pre-specified comparisons reported in Section 4.

3.6 Reproducibility

Table 1 lists all hyperparameters. Code, the full 168-run result table, and the three supplementary datasets described below are provided in the code repository (Declarations).

Table 1 Hyperparameters and implementation details.

Parameter	Value
GCN encoder	2 layers, hidden/embedding dim 24
Node features	degree, clustering coeff., avg. neighbor degree, k -core (min–max normalized)
Optimizer	Adam, learning rate 3×10^{-3}
Discount factor γ	0.98
Entropy coefficient	0.02
Gradient clipping	max norm 1.0
RL rollouts per instance	200 (main study); 80 (N=100 scale-up, Sec. 4.4)
SA temperature schedule	$T_0 = 0.05$, geometric cooling rate 0.995
SA iterations	300 (static objective), 250 (minimax objective)
Deletion fraction α	0.3
Rewiring budget B	15 edges (main study, real-data); 20–30 (scale-up)
Network size N	50 (main study, real-data); 100 (scale-up)
Mean degree (synthetic)	≈ 4
Random seeds	1–8 (main study, per topology); see Sec. 4.4 for supplementary studies
Software	Python 3.12, PyTorch 2.13 (CPU), NetworkX 3.6

3.7 Supplementary validation studies

Three additional, more narrowly scoped studies test the generality of the main finding along axes a reviewer would reasonably question: whether it holds on real infrastructure, whether it depends on the specific three attacks chosen, and whether it survives at larger scale. Each is deliberately smaller than the main grid (fewer seeds, one topology, or a cheaper optimizer) to keep total compute tractable; we report effect directions and, where sample size permits, significance, rather than treating these as independently powered studies.

Real-world infrastructure. We obtained the Western States Power Grid of the United States (Watts & Strogatz, 1998; $N = 4941$, $E = 6594$; MIT-licensed data repository, see Declarations), the same real network on which the original robustness index was validated [8]. Because recalculated betweenness does not scale to $N \sim 5000$ within our compute budget, we sample six connected $N = 50$ subgraphs via breadth-first search from independent random seed nodes, a standard technique for computationally bounded study of large real networks; each sample is a genuine contiguous fragment of the real grid; not a synthetic surrogate, and inherits the grid’s markedly sparser, more tree-like local structure (mean degree ≈ 2.0 – 2.5 versus ≈ 4 for the synthetic study). All seven methods are run on each sample under the same protocol as the main study, evaluated under both the training ($\mathcal{A}$) and expanded ($\mathcal{A}^+$) portfolios.

Attack-portfolio ablation. To test whether all three attacks in $\mathcal{A}$ contribute, or whether one is doing the work, we train SA-Minimax under five portfolio configurations – HDA-only (equivalent to SA-Static), HBA-only, HDA+HBA, HDA+Random, and the full HDA+HBA+Random – on BA topology, 5 seeds, evaluated under $\mathcal{A}^+$. We use SA rather than RL for this and the scale-up study below because Section 4 establishes that the two optimizers are statistically indistinguishable given the same objective, making SA a faithful and ~ 10 – $25\times$ cheaper stand-in that lets us cover more configurations.

Scale-up. To probe whether the effect persists beyond $N = 50$, we repeat a reduced version of the main comparison at $N = 100$ (double the main study’s scale, budget $B = 20$ – 30), 5 seeds, BA topology, using SA-Static and SA-Minimax for all 5 seeds and RL-Static/RL-Minimax as a consistency spot-check on 2 seeds (with reduced rollouts, 80 vs. 200, for tractability, so RL absolute wall-times below are not directly comparable across scale). This also yields matched-scale wall-clock measurements for the complexity discussion in Section 4.4.

4 Results

4.1 Minimax training reduces brittleness without sacrificing robustness

Table 2 summarizes all seven methods pooled across the 24 instances. Every design method improves substantially on the undefended graph. Among static-objective methods, RL-STATIC attains the highest R_{HDA} (0.249 ± 0.040), narrowly ahead of SA-STATIC (0.246 ± 0.040) and clearly ahead of DCBO (0.236 ± 0.043). However, both static-objective methods show a large Adaptive Degradation Gap (RL-STATIC:

0.051 ± 0.028 ; SA-STATIC: 0.049 ± 0.031): more than half of the apparent robustness gain evaporates once the attacker is allowed to switch to a betweenness-based attack on the same topology.

Minimax-objective methods show a markedly different profile. RL-MINIMAX and SA-MINIMAX reduce the Adaptive Degradation Gap to 0.013 ± 0.014 and 0.010 ± 0.010 respectively – a 74–81% reduction relative to their static counterparts (Wilcoxon signed-rank, $n = 24$: $W = 0$ for both comparisons; rank-biserial $r = 1.00$, indicating every one of the 24 paired instances showed a smaller gap under minimax training; Holm-corrected $p < 0.001$ for both; 95% bootstrap CI on the mean paired difference $[0.030, 0.046]$ for RL, $[0.031, 0.050]$ for SA). Figure 3 shows this pattern is consistent across all three topology families, though the absolute gap is largest for WS (small-world) graphs under static training.

Critically, this reduction in brittleness is not a robustness-for-consistency trade: minimax-trained methods also achieve *higher* worst-case robustness R_{worst} than their static counterparts (RL-MINIMAX vs. RL-STATIC: $+0.015$, $r = 0.99$, Holm-corrected $p < 0.001$; SA-MINIMAX vs. SA-STATIC: $+0.018$, $r = 0.94$, Holm-corrected $p < 0.001$; Table 3, Figure 2). Static-objective methods over-fit to the HDA metric they were trained against at the expense of betweenness robustness, and this trade is unfavorable once evaluated honestly: optimizing for one attack leaves the design worse off, not better, under the realistic assumption that an adversary picks its own attack. Full test statistics for all eight pre-specified comparisons are in Table 5.

4.2 Optimizer choice does not explain the effect

Given the same minimax objective, RL-MINIMAX and SA-MINIMAX are statistically indistinguishable on both R_{worst} (mean difference -0.0005 , $r = -0.04$, Holm-corrected $p = 0.88$) and the Adaptive Degradation Gap (mean difference $+0.0035$, $r = 0.23$, Holm-corrected $p = 0.68$). The same near-equivalence holds under the static objective: RL-STATIC does not significantly outperform SA-STATIC on R_{HDA} (Holm-corrected $p = 0.44$), though it does significantly outperform the simpler DCBO greedy heuristic ($r = 0.87$, Holm-corrected $p < 0.001$), confirming the learned policy is functioning correctly rather than trivially. We report this honestly as a central, if less dramatic, finding: at the scale studied here, *the training paradigm accounts for the effect, not the specific optimizer used to pursue it*. A practitioner without deep-learning infrastructure can obtain the same reconnaissance-robustness benefit from simulated annealing, provided the objective itself is minimax rather than single-attack. This finding is what lets us use the cheaper SA optimizer to cover the additional configurations in Section 4.4 without re-litigating the optimizer question there.

4.3 A structural signature: HDA-robust but HBA-fragile

Figure 4 plots R_{HDA} against R_{HBA} for every instance. Static-objective methods cluster visibly below the diagonal, indicating designs that are more robust to degree-based than betweenness-based attack, whereas minimax-objective methods sit close to the diagonal. This is confirmed by which attack the portfolio selects as worst-case (Table 4): for every one of the 96 static-objective-and-baseline instances (RANDOM,

DCBO, SA-STATIC, RL-STATIC), betweenness attack is the most damaging strategy; for minimax-trained methods this drops to 19–21 of 24 instances, with HDA becoming the worst-case attack in a meaningful minority of cases, indicating a genuinely more balanced (though not perfectly symmetric) design. A plausible mechanism is that optimizing purely for R_{HDA} rewards adding edges among already high-degree nodes (directly suppressing the degree-attack ranking) while leaving structurally critical bridging nodes, which betweenness attack targets, comparatively unreinforced.

Table 2 Mean (s.d.) across 24 seed–topology instances, pooled over BA/ER/WS.

Method	R_{HDA}	R_{HBA}	Adaptive Degradation Gap
NoDesign	0.165 (0.036)	0.141 (0.023)	0.024 (0.026)
Random	0.234 (0.037)	0.195 (0.026)	0.039 (0.023)
DCBO	0.236 (0.043)	0.192 (0.030)	0.044 (0.034)
SA-Static	0.246 (0.040)	0.196 (0.027)	0.049 (0.031)
SA-Minimax	0.224 (0.031)	0.214 (0.026)	0.010 (0.010)
RL-Static	0.249 (0.040)	0.198 (0.025)	0.051 (0.028)
RL-Minimax	0.227 (0.034)	0.214 (0.026)	0.013 (0.014)

Table 3 Worst-of-portfolio robustness R_{worst} , mean (s.d.), by topology family.

Method	BA	ER	WS
NoDesign	0.121 (0.011)	0.162 (0.021)	0.139 (0.015)
Random	0.169 (0.011)	0.214 (0.021)	0.203 (0.019)
DCBO	0.170 (0.018)	0.224 (0.025)	0.184 (0.012)
SA-Static	0.170 (0.013)	0.219 (0.027)	0.200 (0.013)
SA-Minimax	0.183 (0.009)	0.228 (0.018)	0.231 (0.008)
RL-Static	0.173 (0.011)	0.223 (0.018)	0.199 (0.012)
RL-Minimax	0.182 (0.011)	0.229 (0.019)	0.229 (0.009)

Table 4 Number of instances (of 24) for which each attack in the portfolio is the worst-case attack.

Method	HBA worst	HDA worst
NoDesign	23	1
Random	24	0
DCBO	24	0
SA-Static	24	0
SA-Minimax	19	5
RL-Static	24	0
RL-Minimax	21	3

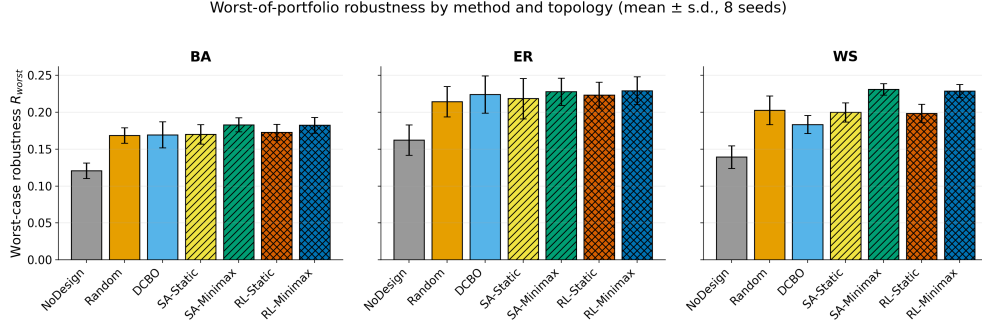


Fig. 2 Worst-of-portfolio robustness R_{worst} by method and topology family (mean $\pm$ s.d., 8 seeds). Minimax-trained methods (dark green, dark orange) match or exceed static-trained methods on the metric a reconnaissance-capable attacker would actually realize.

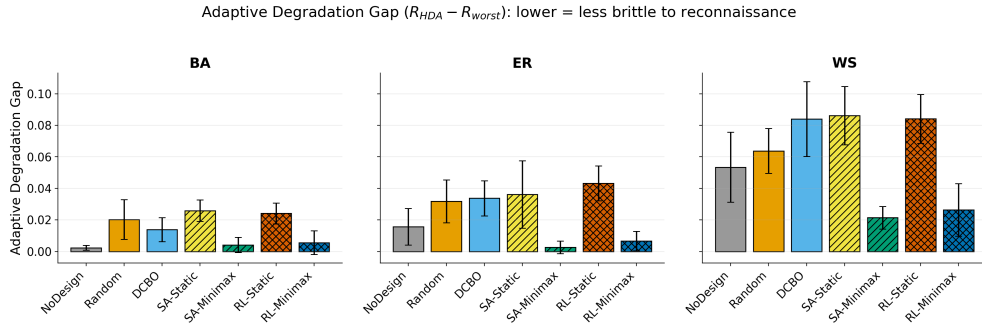


Fig. 3 Adaptive Degradation Gap ($R_{\text{HDA}} - R_{\text{worst}}$) by method and topology family. Lower is less brittle to an attacker that adapts its strategy to the specific defended topology. The gap for minimax-trained methods is 74–81% smaller than for their static-trained counterparts, uniformly across all three topology families.

4.4 Robustness checks: real infrastructure, attack composition, and scale

Real-world power grid. Table 6 reports results on the six BFS-sampled subgraphs of the US Western States Power Grid (Section 3.7). The direction of the main finding replicates: SA-MINIMAX shows a smaller gap than SA-STATIC in all 6 of 6 samples (Wilcoxon $p = 0.031$, the strongest possible signal at $n = 6$) and RL-MINIMAX in 5 of 6 ($p = 0.0625$). Because the expanded portfolio $\mathcal{A}^+$ (adding PageRank, eigenvector, and k -shell attack) changes R_{worst} by less than 0.001 for every method on this sparse, tree-like real topology, we report only the $\mathcal{A}$ -based columns; the near-zero shift itself indicates that training against $\mathcal{A}$ generalizes to the three unseen attacks on this network without measurable loss. With only 6 independent samples this replication is directionally consistent but underpowered relative to the main 24-instance study, and

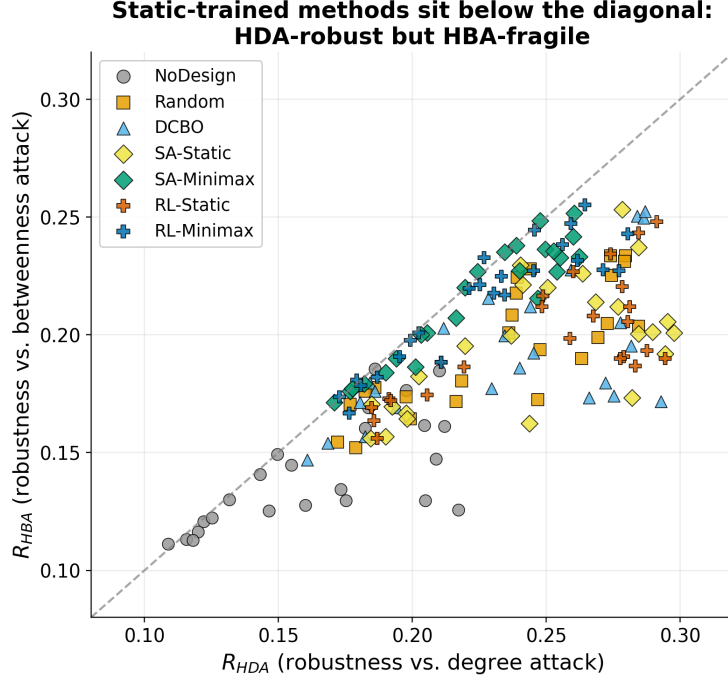


Fig. 4 R_{HDA} versus R_{HBA} for all 168 designed instances. Points below the diagonal are more robust to degree attack than betweenness attack. Static-objective methods (light colors) cluster below the diagonal; minimax-objective methods (dark green, dark orange) sit close to it.

should be read as a real-topology sanity check rather than an independently conclusive result.

Attack-portfolio ablation. Table 7 shows that not all three attacks in the training portfolio contribute equally. HDA+Random performs *identically* to HDA-only ($R_{HDA} = 0.193$, $\text{gap}_6 = 0.021$ for both): because a well-connected network is already fairly tolerant of random failure, random attack essentially never becomes the binding worst-case term, so adding it to the portfolio does nothing. In contrast, HBA-only alone closes most of the gap ($\text{gap}_6 = 0.005$) and nearly matches the full three-attack portfolio ($\text{gap}_6 = 0.003$); HDA+HBA is statistically indistinguishable from the full portfolio. The practical implication is that the effective minimax training portfolio for degree-heterogeneous networks may need only two well-chosen attacks, HDA and HBA, rather than a large or exhaustive set – a more specific and actionable claim than “a portfolio helps,” and a direct answer to why these three attacks were the right starting point rather than, e.g., ten.

Scale-up to $N = 100$. Table 8 repeats a reduced version of the comparison at twice the main study’s network size. The gap-reduction pattern persists in direction for both optimizers (SA: gap reduces from 0.018 ± 0.010 to 0.004 ± 0.003 across all 5 seeds, Wilcoxon $p = 0.0625$, the strongest signal available at this n ; RL: 2/2 seeds show a smaller gap under minimax, too few for a meaningful test statistic but consistent with the SA result and the main study). We treat this as evidence the effect is not an

artefact of the specific $N = 50$ scale, while stopping short of claiming it is established at the hundreds-to-thousands-of-nodes scale common in the wider literature (Section 6).

Runtime and complexity. Table 8 also reports mean wall-clock time per run at both scales. The dominant cost is betweenness centrality, recomputed once per node removal during every R_{HBA} evaluation (Brandes’ algorithm [28], $O(N \cdot E)$ per call, called $O(N)$ times per R_{HBA} evaluation, itself called once per training step); this is why SA-MINIMAX and RL-MINIMAX, which evaluate the minimax objective at every step, cost roughly $20\times$ and $7\times$ their static counterparts respectively, and why doubling N increased SA-MINIMAX wall-time by $\approx 4\times$. This cost, not the training paradigm itself, is the binding constraint on scaling the present approach to the thousands-of-nodes networks common in the literature we build on, and is the most direct target for future engineering work (e.g. incremental or sampled betweenness approximations).

Table 5 Full paired-test statistics for the eight comparisons reported in Section 4: mean paired difference, 95% bootstrap CI, matched-pairs rank-biserial effect size r , and Holm–Bonferroni-corrected p -value ($n = 24$ throughout).

Comparison (metric)	Mean diff.	95% CI	r	p_{Holm}
RL-Static vs. RL-Minimax (gap)	+0.038	[0.030, 0.046]	1.00	< 0.001
SA-Static vs. SA-Minimax (gap)	+0.040	[0.031, 0.050]	1.00	< 0.001
RL-Minimax vs. RL-Static (R_{worst})	+0.015	[0.010, 0.021]	0.99	< 0.001
SA-Minimax vs. SA-Static (R_{worst})	+0.018	[0.012, 0.024]	0.94	< 0.001
RL-Static vs. DCBO (R_{HDA})	+0.013	[0.007, 0.018]	0.87	< 0.001
RL-Minimax vs. SA-Minimax (R_{worst})	−0.0005	[−0.003, 0.002]	−0.04	0.88
RL-Minimax vs. SA-Minimax (gap)	+0.0035	[−0.002, 0.009]	0.23	0.68
RL-Static vs. SA-Static (R_{HDA})	+0.0031	[−0.001, 0.007]	0.34	0.44

Table 6 Real-world validation: US Western States Power Grid, 6 BFS-sampled $N=50$ connected subgraphs, mean (s.d.).

Method	R_{HDA}	R_{worst}	Adaptive Degradation Gap
NoDesign	0.064 (0.010)	0.064 (0.009)	0.001 (0.001)
DCBO	0.098 (0.018)	0.092 (0.015)	0.006 (0.009)
SA-Static	0.135 (0.022)	0.116 (0.015)	0.019 (0.010)
SA-Minimax	0.122 (0.017)	0.120 (0.016)	0.003 (0.004)
RL-Static	0.137 (0.024)	0.116 (0.018)	0.021 (0.013)
RL-Minimax	0.128 (0.024)	0.121 (0.016)	0.007 (0.009)

5 Discussion

The central finding is that the training *objective* used to design a robust network topology matters more than which optimizer pursues it, and that the field’s default objective, robustness against one attack fixed in advance, produces a systematic and previously undocumented failure mode: brittleness to any attack outside the one

Table 7 Attack-portfolio ablation (SA-based, BA topology, 5 seeds), evaluated under the expanded 6-attack portfolio $\mathcal{A}^+$.

Training portfolio	R_{HDA}	R_{HBA}	R_{worst}^+	Gap ⁺
HDA-only (=SA-Static)	0.193 (0.006)	0.171 (0.010)	0.171 (0.010)	0.021 (0.008)
HDA+Random	0.193 (0.006)	0.171 (0.010)	0.171 (0.010)	0.021 (0.008)
HBA-only	0.183 (0.008)	0.181 (0.006)	0.179 (0.003)	0.005 (0.005)
HDA+HBA	0.186 (0.008)	0.183 (0.006)	0.183 (0.006)	0.003 (0.002)
HDA+HBA+Random (full)	0.184 (0.008)	0.181 (0.006)	0.181 (0.006)	0.003 (0.002)

Table 8 Scale-up to $N=100$ (BA topology) and mean wall-clock time per run at both scales.

Method	Gap ($N=50$)	Gap ($N=100$)	Time, $N=50$ (s)	Time, $N=100$ (s)
DCBO	0.044 (0.034)	0.009 (0.007)	0.10	0.35
SA-Static	0.049 (0.031)	0.018 (0.010)	0.50	1.65
SA-Minimax	0.010 (0.010)	0.004 (0.003)	9.68	39.10
RL-Static	0.051 (0.028)	0.025 (0.004) ¹	17.25	29.72 ¹
RL-Minimax	0.013 (0.014)	0.005 (0.001) ¹	123.29	247.48 ¹

¹ $N=100$ RL rows use 2 seeds and 80 rollouts (vs. 200 at $N=50$) for tractability; not directly comparable in absolute wall-time to the $N=50$ column.

trained against, most consistently betweenness-based attack. This is not a minor effect at the margins; it erases roughly half of the apparent robustness gain reported by the standard evaluation protocol once a reconnaissance-capable adversary is assumed, and it holds identically whether the optimizer is a deep GNN policy or classical simulated annealing.

This has a direct practical implication for anyone applying learned or heuristic network-robustness-design tools to real infrastructure: reporting robustness against the single attack a method was optimized for, the norm in the literature we build on [10–12], systematically overstates real-world robustness against an adversary who is not obliged to attack in the way the designer assumed. The fix demonstrated here is inexpensive: replace the fixed attack objective with a worst-of-small-portfolio objective, recomputed during training, at negligible additional implementation cost and only a moderate increase in per-step evaluation cost.

The null result on optimizer choice is, we think, equally informative. It tempers a natural assumption that the growing literature on GNN-based network design owes its gains primarily to representational power. In our setting, at moderate scale, a 1980s-vintage local-search method matches a modern graph neural network once both pursue the same objective. This does not contradict prior work showing RL outperforming heuristics at larger scale or when *generalizing* across a distribution of graphs without retraining [10–12] – our instance-specific setting removes exactly that advantage (Section 6) – but it does suggest that objective design deserves at least as much attention as architecture design in this literature.

The three supplementary checks (Section 4.4) narrow, though do not eliminate, three natural objections to the main result. First, the effect is not an artefact of synthetic graphs: it replicates directionally on genuine fragments of a real power grid,

whose sparse, near-tree-like local structure is qualitatively different from any of the three synthetic families. Second, it is not merely “more attacks are better”: the ablation shows random attack contributes nothing to the training portfolio, and that HDA and HBA together, or even HBA alone, capture nearly all of the benefit, which is a more falsifiable and more actionable claim than portfolio size. Third, the effect persists in direction at double the network scale, though our compute budget could not establish this with the statistical power of the main study. We regard the mechanism identified here, that the attacks a design is *not* evaluated against during training are exactly the attacks it becomes vulnerable to, as more fundamental than any of the three specific instantiations (RL, SA, or the particular attack set) used to demonstrate it, which is why we expect it to hold beyond the scope tested directly.

6 Limitations

We report scope choices plainly rather than understating them.

Scale. The main study’s $N = 50$ networks are small relative to the $N = 100$ –5000 range in [10, 12]. Section 4.4 shows the effect persists at $N = 100$, but with only 5 seeds (SA) or 2 (RL) on one topology – well below the main grid’s power, and still an order of magnitude below the literature’s upper range. Table 8 shows the obstacle is not the training paradigm but recalculated betweenness cost ($4\times$ wall-clock for a $2\times$ increase in N); scaling further needs an approximate or incremental betweenness estimator, not more compute.

Instance-specific optimization. Unlike RNet-DQN, ResiNet, and Zhu et al., which train one inductive policy generalizing across graph instances, we optimize a fresh policy per instance, matching the SA baseline’s regime for a fair comparison. This isolates the objective effect cleanly but leaves open whether an inductive minimax policy generalizes zero-shot to unseen graphs – true of the real-data, ablation, and scale-up checks too. We consider this the most important direction for follow-up work.

Attacker model. Our minimax attacker selects among a small, cheaply computable strategy set rather than a fully trained adversary as in RARL’s self-play [13]. The ablation shows two attacks (HDA, HBA) already capture nearly all the benefit, narrowing the practical need for a larger portfolio – but a learned adversary could still find damage strategies outside any fixed, human-specified set. A genuinely co-trained, alternating-optimization attacker [13] is the natural next step, and the one Zhu et al. identify as open [12].

Real-world validation. Section 4.4 confirms the effect’s direction on real power-grid subgraphs. Two limits remain: BFS-sampling to $N = 50$ rather than validating the full network (infeasible for the same betweenness-cost reasons as the scale-up study), and $n = 6$ samples, well below the main study’s power. This is a real but modest replication, not a substitute for a larger real-network study.

Optimizer implementation. We use single-trajectory A2C rather than clipped-objective PPO, for implementation reliability within budget. We would not expect this to change the optimizer-invariance finding, since SA – an entirely different, non-neural method – shows the identical pattern, but it may understate the absolute performance a more tuned RL implementation could reach.

Nature of the contribution. We note explicitly, rather than leaving it implicit, that this paper’s contribution is conceptual and empirical rather than architectural: we introduce no new network architecture, no new base optimization algorithm, and no new GNN layer. The contribution is the identification of a previously undocumented failure mode in an established paradigm, a lightweight and general fix (reformulating the training objective as a minimax over a small, well-chosen attack portfolio), and the empirical demonstration, across two optimizers, three synthetic topology families, one real network, and two scales, that the fix works and that objective design matters at least as much as the optimizer used to pursue it. We consider this a legitimate and, we would argue, undersupplied kind of contribution in a literature that has concentrated heavily on architectural novelty.

7 Conclusion

Training network-robustness-design agents against a single, fixed attack strategy, the near-universal practice in both classical and learned approaches to this problem, produces designs that are quietly brittle to an adversary who reconnoiters and adapts. A lightweight minimax training objective, evaluating the worst of a small attack portfolio at every training step rather than one attack chosen in advance, closes most of this gap and does so without sacrificing raw worst-case robustness, across three canonical topology families and two independent optimizers, with the effect direction confirmed on a real power grid and at double the network scale. The training objective, not the optimizer, is the operative ingredient, and a targeted ablation shows the effect is driven specifically by including betweenness-based attack in the training portfolio rather than by portfolio size per se. We see the natural extensions as scaling the attacker portfolio to a fully learned, co-trained adversary, testing whether an inductive (rather than instance-specific) minimax policy generalizes zero-shot to unseen graphs, and validating the approach on full-scale real infrastructure networks once betweenness recomputation can be made to scale to that regime.

Declarations

- **Funding:** This research received no external funding.
- **Conflict of interest:** The author declares no competing interests.
- **Ethics approval:** Not applicable; this study involves no human or animal subjects.
- **Consent to participate:** Not applicable.
- **Consent for publication:** Not applicable.
- **Availability of data and materials:** All synthetic networks are generated from documented generative models (Barabási–Albert, Erdős–Rényi, Watts–Strogatz) with fixed random seeds. The real-world validation uses the Western States Power Grid of the United States (Watts & Strogatz, 1998), obtained from the Critical-Infrastructure-Networks data repository (Zuev, K.M., <https://github.com/Kostia-Zuev/Critical-Infrastructure-Networks>, MIT license, DOI: 10.5281/zenodo.583631). The full result tables for the main study (168 runs), real-data validation, attack-portfolio ablation, and scale-up study are available in the code repository below.

- **Code availability:** <https://github.com/hasib1233333/Minimax-Training-Closes-the-.git>
- **Authors' contributions:** M.H. conceived the study, designed and implemented the simulation framework, ran all experiments, performed the statistical analysis, and drafted the manuscript.

References

- [1] Newman, M.E.J.: The structure and function of complex networks. *SIAM Review* **45**(2), 167–256 (2003) <https://doi.org/10.1137/S003614450342480>
- [2] Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. *Reviews of Modern Physics* **74**(1), 47–97 (2002) <https://doi.org/10.1103/RevModPhys.74.47>
- [3] Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.-U.: Complex networks: Structure and dynamics. *Physics Reports* **424**(4-5), 175–308 (2006) <https://doi.org/10.1016/j.physrep.2005.10.009>
- [4] Albert, R., Jeong, H., Barabási, A.-L.: Error and attack tolerance of complex networks. *Nature* **406**(6794), 378–382 (2000) <https://doi.org/10.1038/35019019>
- [5] Holme, P., Kim, B.J., Yoon, C.N., Han, S.K.: Attack vulnerability of complex networks. *Physical Review E* **65**, 056109 (2002) <https://doi.org/10.1103/PhysRevE.65.056109>
- [6] Motter, A.E., Lai, Y.-C.: Cascade-based attacks on complex networks. *Physical Review E* **66**, 065102 (2002) <https://doi.org/10.1103/PhysRevE.66.065102>
- [7] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983) <https://doi.org/10.1126/science.220.4598.671>
- [8] Schneider, C.M., Moreira, A.A., Andrade, J.S., Havlin, S., Herrmann, H.J.: Mitigation of malicious attacks on networks. *Proceedings of the National Academy of Sciences* **108**(10), 3838–3841 (2011) <https://doi.org/10.1073/pnas.1009440108>
- [9] Mnih, V., *et al.*: Human-level control through deep reinforcement learning. *Nature* **518**(7540), 529–533 (2015) <https://doi.org/10.1038/nature14236>
- [10] Darvari, V.-A., Hailes, S., Musolesi, M.: Goal-directed graph construction using reinforcement learning. *Proceedings of the Royal Society A* **477**(2254), 20210168 (2021) <https://doi.org/10.1098/rspa.2021.0168>
- [11] Yang, S., Ma, K., Wang, B., Yu, T., Zha, H.: Learning to boost resilience of complex networks via neural edge rewiring. *Transactions on Machine Learning Research* (2023)

- [12] Zhu, B., Zhu, T., Gao, J., Havlin, S., others, Li, D.: Design of robust networks via reinforcement learning prompts the emergence of multi-backbones. *Nature Communications* **17**, 4278 (2026) <https://doi.org/10.1038/s41467-026-70745-0>
- [13] Pinto, L., Davidson, J., Sukthankar, R., Gupta, A.: Robust adversarial reinforcement learning. In: *Proceedings of the 34th International Conference on Machine Learning*, pp. 2817–2826 (2017)
- [14] Callaway, D.S., Newman, M.E.J., Strogatz, S.H., Watts, D.J.: Network robustness and fragility: Percolation on random graphs. *Physical Review Letters* **85**, 5468–5471 (2000) <https://doi.org/10.1103/PhysRevLett.85.5468>
- [15] Cohen, R., Havlin, S., ben-Avraham, D.: Efficient immunization strategies for computer networks and populations. *Physical Review Letters* **91**, 247901 (2003) <https://doi.org/10.1103/PhysRevLett.91.247901>
- [16] Alenazi, M.J.F., Cetinkaya, E.K., Sterbenz, J.P.G.: Cost-efficient algebraic connectivity optimisation of backbone networks. *Optical Switching and Networking* **14**, 107–116 (2014) <https://doi.org/10.1016/j.osn.2014.05.011>
- [17] Freeman, L.C.: A set of measures of centrality based on betweenness. *Sociometry* **40**(1), 35–41 (1977) <https://doi.org/10.2307/3033543>
- [18] Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems* **30**(1-7), 107–117 (1998) [https://doi.org/10.1016/S0169-7552\(98\)00110-X](https://doi.org/10.1016/S0169-7552(98)00110-X)
- [19] Bonacich, P.: Power and centrality: A family of measures. *American Journal of Sociology* **92**(5), 1170–1182 (1987) <https://doi.org/10.1086/228631>
- [20] Kitsak, M., Gallos, L.K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H.E., Makse, H.A.: Identification of influential spreaders in complex networks. *Nature Physics* **6**(11), 888–893 (2010) <https://doi.org/10.1038/nphys1746>
- [21] Morone, F., Makse, H.A.: Influence maximization in complex networks through optimal percolation. *Nature* **524**(7563), 65–68 (2015) <https://doi.org/10.1038/nature14604>
- [22] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations (ICLR)* (2017)
- [23] Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T.P., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1928–1937 (2016)

- [24] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
- [25] Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999) <https://doi.org/10.1126/science.286.5439.509>
- [26] Erdős, P., Rényi, A.: On random graphs i. *Publicationes Mathematicae Debrecen* **6**, 290–297 (1959)
- [27] Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. *Nature* **393**(6684), 440–442 (1998) <https://doi.org/10.1038/30918>
- [28] Brandes, U.: A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology* **25**(2), 163–177 (2001) <https://doi.org/10.1080/0022250X.2001.9990249>