Appendix

1 Active Search

We plot the robustness improvement (RI) against targeted (random) attack as a function of training steps in Fig. 1 (Fig. 2). The x-axis represents number of training steps and the y-axis represents the RIs obtained with the corresponding training steps. As we can see from Fig. 1, for targeted attack, NEP-AM (NEP-HAM) trained for 10k steps can achieve the best RIs (the RIs obtained by training NEP-AM (NEP-HAM) 50k steps) on 7 (6) out of 10 networks. All results of NEP-AM (NEP-HAM) obtained in 10k steps are already significantly better than the results of NEP-DQN. We perform a similar analysis for Fig. 2. For random attack, as shown in Fig. 2, NEP-AM (NEP-HAM) trained for 20k steps can achieve the best RIs on 6 out of 10 networks. The results of NEP-AM (NEP-HAM) obtained in 20k steps are better or comparable (the relative differences are less than about 0.4%) than that of NEP-DQN on all networks, except for CZ and GB, where the best results of NEP-AM (NEP-HAM) are slightly worse than NEP-DQN.

2 Selection of Features

To improve action and state representations, both relying on the node representations, we construct node features to provide high-quality node representations. Researchers in the filed of complex networks have defined a large number of measurements to evaluate node centrality, which have the ability to distinguish nodes. However, some of the centrality measures such as closeness [Freeman and others, 2002] and betweenness [Freeman, 1977] do not meet our requirements in that they are computationally expensive. We choose the following two centrality measures because they take into account the local neighborhood information and are very easy to calculate.

1. (Degree) The degree is a simple, yet discriminative, feature of a node. Networks show a variation in node degree to some extent. For instance, the node degree follows a binomial distribution in ER networks. BA networks and many real networks, whose node degrees follow a power law distribution, have a larger variation in node degree [Sánchez Martínez, 2009].

2. (Average Neighbor Degree) To distinguish node u and v with the same degree, we compute their average neighbor degree as below to add extra information.

$$d_n(u) = \frac{1}{|\mathcal{N}(u)|} \sum_{w \in \mathcal{N}(u)} d(w) \tag{1}$$

where $\mathcal{N}(u)$ is the set of first-order neighbors, $d(w)$ is the degree of node w.

As a complement to these centrality measures, we introduce cluster coefficient to compute the density of edges in the node's local neighborhood. The cluster coefficient of node u is defined as below:

$$c(u) = T(u)/\binom{d(u)}{2} \tag{2}$$

where $T(u)$ is the number of triangles in the neighborhood of u. In the implementation, we dynamically adjust the cluster coefficient to realize its efficient computation in the decision-making process.

In addition to topological node features, the relative relationships between the edge stub and all other nodes are introduced to enhance node features. Motivated by decision criteria of heuristic algorithms [Wang and Van Mieghem, 2010; Li *et al.*, 2018], we compute the distance, product of degree, and algebraic distance between the edge stub and all other nodes. The last measure introduced (Jaccard coefficient) characterizes the proximity between the edge stub and all other nodes, which is defined as follow.

$$J(u,v) = \frac{|\mathcal{N}(u) \cap \mathcal{N}(v)|}{|\mathcal{N}(u) \cup \mathcal{N}(v)|} \tag{3}$$

As we mentioned in the text, we also embed all nodes into the two-dimensional Euclidean space to reflect the global position of nodes in the network. All features are normalized to $[0, 1]$. We show the distribution of node features, node coordinates for a BA (ER) network in Fig. 3 (Fig. 4). For each feature, we observe a variation in feature values across nodes in the network. We combine all features to learn high-quality node representations.

3 Run Times

Comparing run times can be challenging due to differences in implementation and hardware. To provide a reference

Table 1: Run Times. The unit is the second.

Network Model	Edge Budget	LDP	FV	ERes	NEP-DQN	NEP-AM	NEP-HAM
BA-500	250	3994	3973.12	10673.54	244.595	735.394	734.118
ER-500	250	3906.816	4006.912	10923.26	1090.386	736.743	746.968

for readers, we list the run times of all baselines and NEP-AM/HAM on 128 synthetic networks of size 500 in Table 1[1]. As we can see from Table 1, the classic heuristics take much longer than learned policies. The most time-consuming part of heuristics is computing the decision criteria for all non-connected node pairs, where the parallel computation of GPU is unavailable. Surprisingly, LDP, which has a complexity of $O(n^2)$, takes a similar amount of time as FV, which has a complexity of $O(n^3)$ and involves computing the eigendecomposition. However, we can improve the efficiency of LDP by adopting more sophisticated data structures. ERes takes longer than FV since it involves computing the pseudoinverse of the graph Laplacian. NEP-DQN is more efficient than our models on BA-500, for it does not need to construct features like our models. However, NEP-DQN takes longer time than our models on ER-500 networks. The reason is that ER-500 networks are denser than BA-500 networks, so the multiple GNN layers in the NEP-DQN model obviously increase its running time.

References

[Freeman and others, 2002] Linton C Freeman et al. Centrality in social networks: Conceptual clarification. *Social network: critical concepts in sociology. Londres: Routledge*, 1:238–263, 2002.

[Freeman, 1977] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[Li *et al.*, 2018] Gang Li, Zhi Feng Hao, Han Huang, and Hang Wei. Maximizing algebraic connectivity via minimum degree and maximum distance. *IEEE Access*, 6:41249–41255, 2018.

[Sánchez Martínez, 2009] Mari Carmen Sánchez Martínez. Robustness optimization via link additions. Master's thesis, Universitat Politècnica de Catalunya, 2009.

[Wang and Van Mieghem, 2010] Huijuan Wang and Piet Van Mieghem. Algebraic connectivity optimization via link addition. In *3d International ICST Conference on Bio-Inspired Models of Network, Information, and Computing Systems*, 2010.

[1]We use the CPU version of NEP-DQN because its GPU version is not compatible with our hardware. If we can run NEP-DQN on GPU, it should be faster.

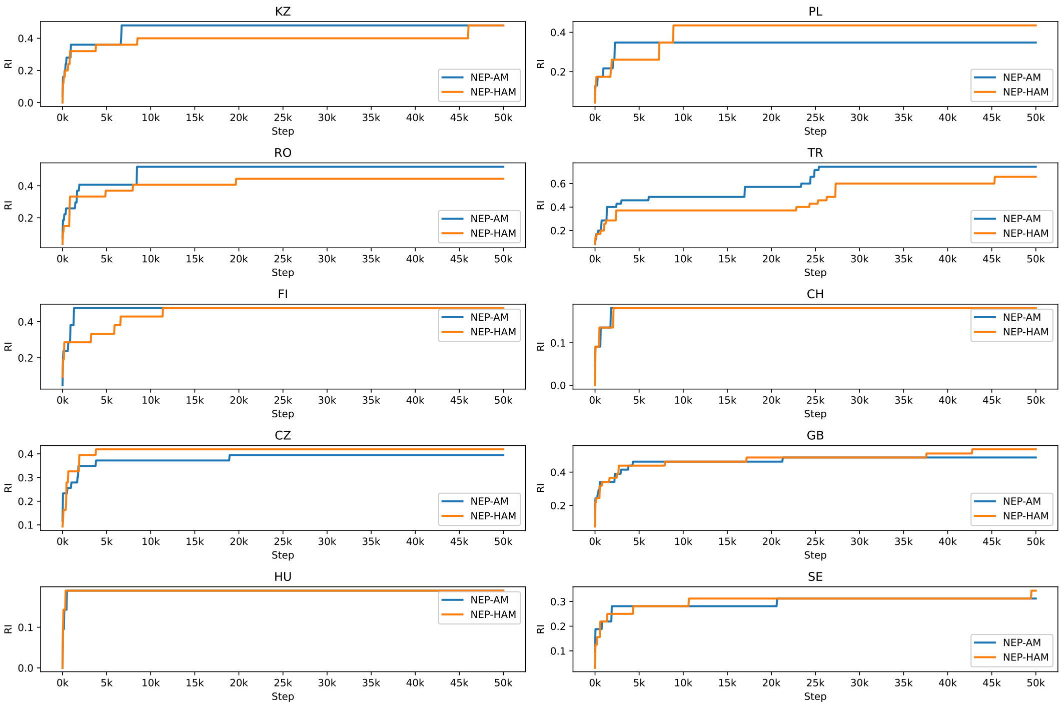

Figure 1: Active search on real networks. The robustness improvement is measured against the targeted attack.

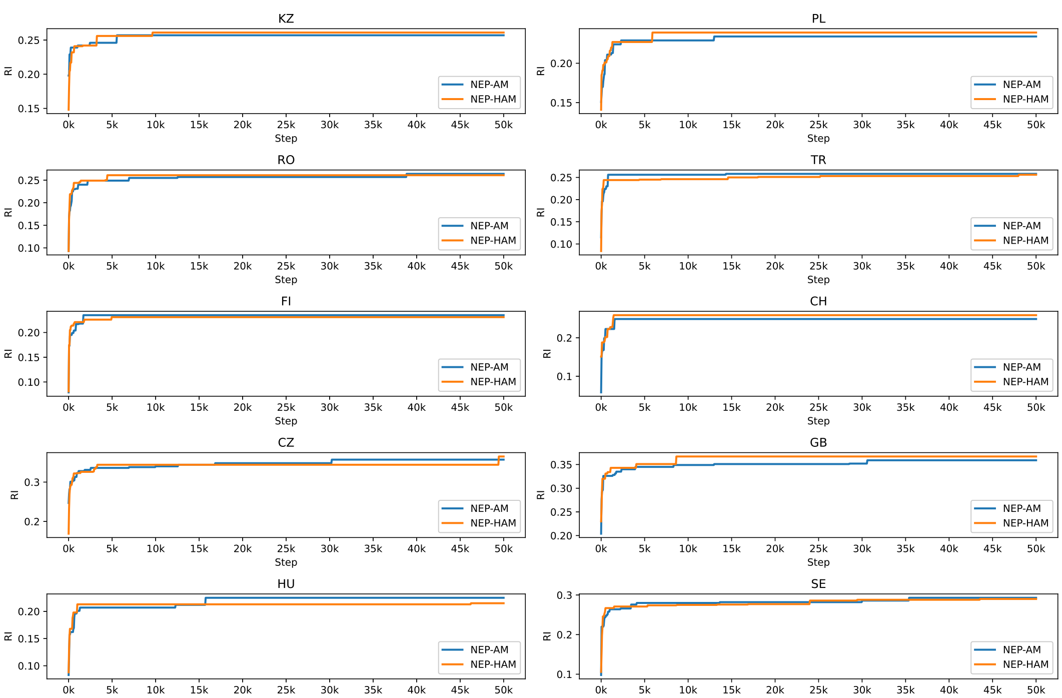

Figure 2: Active search on real networks. The robustness improvement is measured against the random attack.

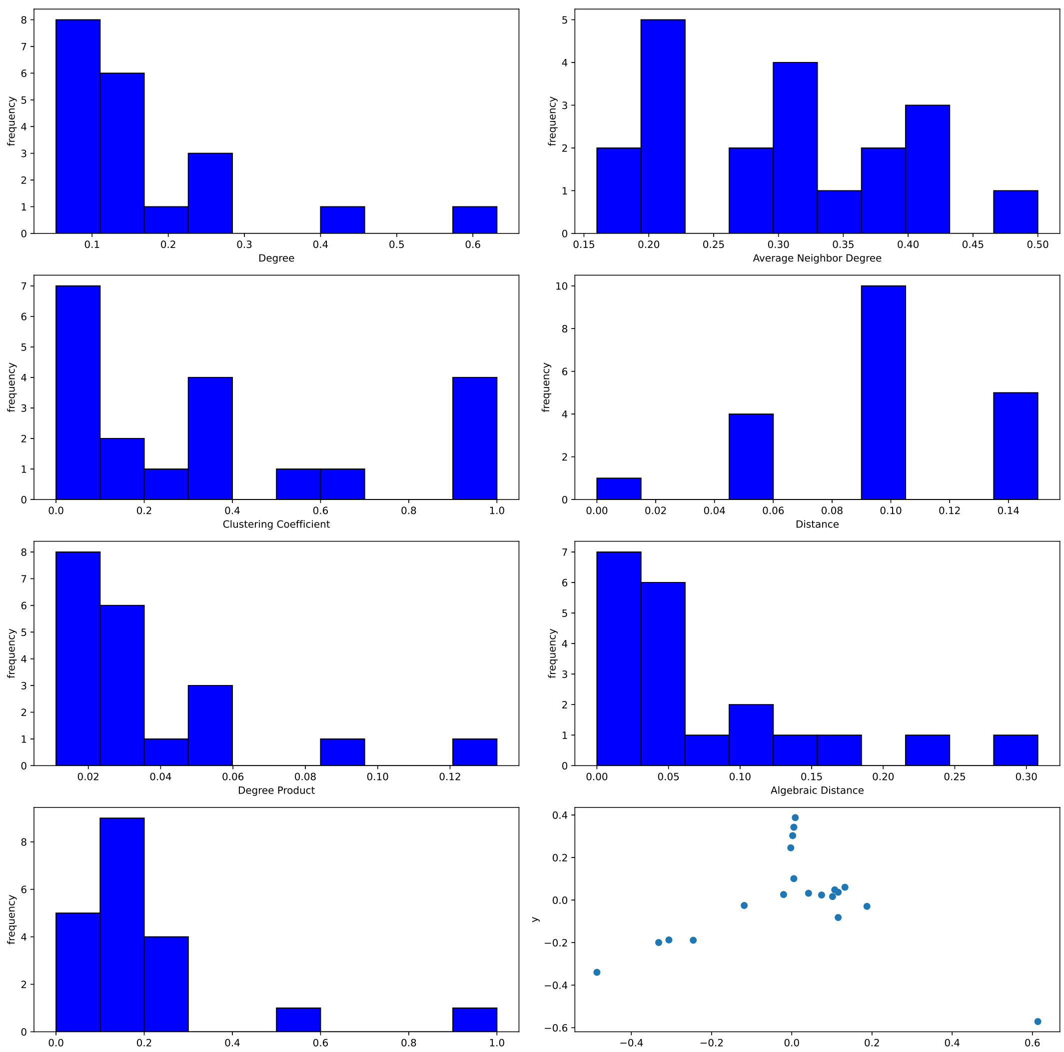

Figure 3: Node features of a BA network.

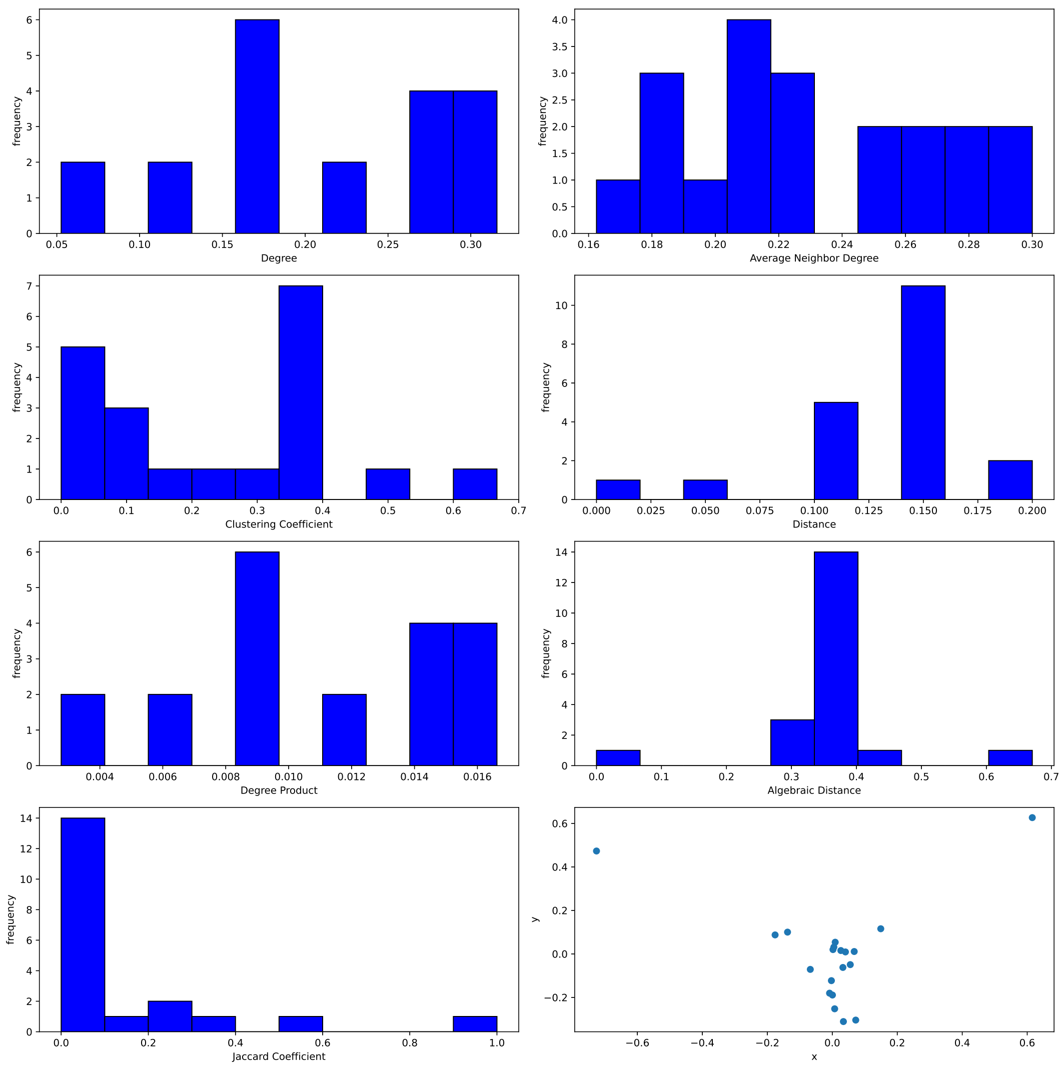

Figure 4: Node features of an ER network.