Discovering network resilience formulas with symbolized reinforcement learning

Yu Zheng[1,2,3], Jingtao Ding[4,*], Depeng Jin[1,2], Jianxi Gao[5,6], and Yong Li[1,2,*]

[1]Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China.
[2]Department of Electronic Engineering, Tsinghua University, Beijing, China.
[3]Massachusetts Institute of Technology, Cambridge, MA, USA.
[4]Department of Earth System Science, Tsinghua University, Beijing, China.
[5]Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, USA.
[6]Network Science and Technology Center, Rensselaer Polytechnic Institute, Troy, NY, USA.
[*]corresponding author: dingjt15@tsinghua.org.cn; liyong07@tsinghua.edu.cn

Many complex networks display remarkable resilience under external perturbations, internal failures and environmental changes, yet they can swiftly deteriorate into dysfunction upon the removal of a few keystone nodes. Discovering formulas that measure network resilience offers the potential to prevent catastrophic collapses—from species extinctions to financial crises—with profound implications for real-world systems. Existing resilience metrics are typically designed to characterize topological properties, whereas in many real systems resilience is defined by the dynamics imposed on the network; consequently, metrics that explicitly incorporate dynamical state information are expected to be better suited for identifying keystone nodes in such settings. Here, we report an automatic method for resilience metric discovery, which learns from how machine learning model solves a complicated network dismantling problem and symbolizes its network attack strategies into mathematical formulas. This proposed self-inductive approach discovers a resilience formula that accounts for both topology and dynamics, highlighting how the correlation between node degree and state shapes overall network resilience. Additionally, our approach discovers formulas that refine existing well-established resilience metrics with over 37.5% improvement in accuracy, significantly advancing human understanding of complex networks with machine learning.

Introduction

Resilience, an essential property of complex networks, describes their ability to maintain basic functionality when internal failures and external disturbances occur[1–3]. The loss of resilience can lead to catastrophic collapses, often caused by the removal of a few keystone nodes vital to the system's functionality[3–5]. For example, the malfunction of a few hub nodes can fragment a communication network, disabling its global information-carrying capacity[3]. Similarly, knocking out a few genes might suppress the expression of the entire cellular network, leading to cell death[2]. Given the pervasive presence of complex networks and the frequent occurrence of perturbations that compromise resilience in real-world systems, it is crucial to understand how network structure and the underlying dynamics influence their overall system resilience[6].

In a long-standing effort, physicists have developed a series of network resilience metrics to identify keystone nodes whose removal triggers the fastest loss of resilience, aiming to rigorously assess network resilience[2,3,6]. These approaches[6]—ranging from manually derived formulas including degree centrality[3] (DC: d_i), collective influence[7] (CI: $(d_i-1)\sum_{j\in\partial\,\mathrm{Ball}(i,l)}(d_j-1)$), and resilience centrality[2,8] (RC: $2\bar{d}_i+d_i(d_i-2\beta)$) to recently emerged data-driven machine learning methods[4,9]—predominantly focus on the topological structure of networks, while leaving a significant gap in addressing the dynamics that drive system functionality. Here d_i and $\bar{d}_i$ denote the degree and the average neighbor degree of node i. $\mathrm{Ball}(i,l)$ is the set of nodes inside a ball of radius l (defined as the shortest path) around node i, and $\partial\,\mathrm{Ball}(i,l)$ is the frontier of the ball. $\beta=\langle d\rangle+\frac{\sigma^2}{\langle d\rangle}$, where $\langle d\rangle$ and σ^2 are the expectation and variance of the degree distribution. Due to the complex coupling of network topology and dynamics, discovering resilience metrics is challenging, exceeding the limits of current analytical approaches that often oversimplify or overlook network dynamics. Moreover, the vast solution space involved in identifying keystone nodes from large networks and deriving mathematical formulas requires efficient search capabilities, rendering current approaches intractable.

Machine learning, emerging as an indispensable tool for addressing challenges beyond human capability, holds vast untapped potential to achieve symbolic discoveries by solving complex problems. In this work, we explore the possibility in discovering network resilience formulas. Specifically, we use reinforcement learning to automatically reveal insightful resilience indicators and derive principled topological and dynamical characteristics that identify keystone nodes. Here, resilience metric discovery is embedded into a network dismantling process where the objective is to induce a network collapse by removing the minimal number of nodes. The designed symbolized reinforcement learning approach not only achieves accurate network

dismantling but also learns from how it solves this complicated problem, deriving network resilience formulas in a self-inductive manner (see Figure 1a). To address the enormous solution space, we develop a deep reinforcement learning (DRL) agent that utilizes neural networks to guide the search of keystone nodes as attack targets. The agent is then symbolized into mathematical formulas revealing its node selection strategy. The discovered resilience formulas can be directly applied to identify keystone nodes and, more importantly, to characterize and quantify network resilience. Unlike previous metrics[2,3,8] that focus solely on topological robustness, our approach derives a resilience formula that integrates both topology and dynamics, expressed as $d \cdot s$, the product of degree and state. This dynamical resilience metric provides a concise yet precise measure of individual nodes' contributions to system resilience, surpassing current network robustness metrics by over 74.1% in accurately identifying keystone nodes.

Our discovered formula also leads to a network-level resilience indicator, the correlation between degree and state distributions, $\mathrm{corr}(d,s)$, which serves as a potential indicator of resilience in many complex networks, even for networks previously considered vulnerable to attacks[3]. The effectiveness of this resilience indicator is empirically verified in real-world ecology and supply chain networks, offering a perspective that explains why natural systems are more resilient to perturbations than man-made ones[10]. Whereas existing data-driven machine learning approaches output keystone nodes directly with black-box neural networks, our approach instead establishes a framework that discovers symbolic resilience metrics using machine learning.

Results

Method Overview

Identifying keystone nodes that determine the resilience of a network is challenging both theoretically and computationally. Theoretically, the resilience function resides on manifolds within a high-dimensional space defined by the complex coupling of network topology and non-linear, heterogeneous system dynamics, which is often non-analytic. Computationally, selecting a subset of nodes from a large network is well-known to be an NP-hard problem[11], with an enormous solution space of size $\frac{N!}{K!(N-K)!}$, which grows super-exponentially as the network size N and the number of selected nodes K increase. For instance, choosing 50 nodes from a 500-node network results in over 10^{133} possible combinations, rendering exhaustive search impractical.

We propose to identify keystone nodes recursively with a DRL agent, which removes one node at each step until the network loses resilience (see Figure 1b and Section M.1-M.3). Specifically, we design a graph neural network (GNN) tailored to encode rich features of both network topology and complex dynamics into node representations, effectively capturing the high-dimensional, non-linear, and heterogeneous resilience function. Informed by the learned node representations, the agent calculates the probability of selecting different nodes and estimates the system resilience with two multi-layer neural networks, i.e. policy network and value network (Figure 1b). This approach enables the agent to strategically select nodes at each step, efficiently exploring the enormous solution space by prioritizing high-quality actions. Therefore, we effectively address the NP-hardness of the problem by searching the minimal keystone nodes whose removal brings the fastest loss of network resilience, rather than exhaustively enumerating all possible solutions. We train the agent for 1 million steps of node removal samples across various networks, and eventually, the agent becomes an expert in identifying keystone nodes for the resilience. The agent outperforms both current formulas[2,3,7,8] and machine learning based approaches[4,9] thanks to its efficient search of the solution space and its comprehensive integration of both network topology and dynamics.

Network metrics are often expressed by mathematical formulas that combine physical variables with operators, but the DRL agent functions as a black-box neural network containing thousands of parameters that, while optimized, lack direct physical meanings, which is difficult to understand by humans. Towards this end, we symbolize the DRL agent's behavior to discover a human-comprehensible mathematical formula that effectively approximates its node-selection strategy, translating the emergent behavior of the black box into an interpretable heuristic.

The rich input topological and dynamical features and feasible mathematical operators result in an infinite number of possible formulas that could approximate the DRL agent, rendering formula discovery challenging. To narrow down the search space, we distill the DRL agent to focus only on its key ingredients by attributing its prediction to individual input features (see Figure 1c and Section M.3). We achieve this by learning a differentiable feature selection mask that maximizes the mutual information between the selected features and the model's prediction while simultaneously minimizing the number of included features. This process results in a sparse set of important features that play dominant roles in the agent's prediction. The distillation effectively simplifies the DRL agent and allows for formula discovery within a streamlined subspace of mathematical formulas, considering only the most informative features.

To generate human-understandable and physically insightful formulas, we develop an evolutionary algorithm to explore the enormous mathematical expression space, mapping the important features to identified keystone nodes towards higher regression accuracy and lower expression complexity, rather than perfectly fitting the data which can lead to over-complicated

formulas (see Figure 1d and Section M.5-M.6). We design an iterative refinement strategy that repeats the evolutionary process with a decreasing set of operators, progressively narrowing the search space based on obtained results. Ultimately, this process generates candidate mathematical formulas that effectively approximate the DRL agent's assessment of node importance, transforming the black-box model's behavior into a set of interpretable heuristics. We involve human experts to pick the most reasonable formulas from the top performed candidates[12], where reasonable refers to formulas that possess physical significance or can be explained by established metrics[2].

Conceptually, as illustrated in Figure 1a, our approach introduces a self-inductive framework—first solve the complex network dismantling problem using DRL, and then learn from how DRL solves the problem by symbolizing its node selection strategies into resilience metrics. The discovered formulas can be directly applied to identify keystone nodes and predict network resilience. Now, we demonstrate how this framework has successfully discovered resilience metrics facing real high-dimensional and non-linear systems that were previously considered intractable with current analytic tools.

Resilience metric accounting for both topology and dynamics

Existing formulas[2,3,5,7,8,13] only reveal the relationship between resilience and network topology, disregarding system dynamics. However, the interplay between topology and dynamics is pervasive in real-world systems, yet its complexity exceeds beyond the capacity of current analytical tools[2], leaving resilience formulas that account for both aspects still unknown. We begin with a typical example of widely investigated gene regulatory networks driven by Michaelis–Menten dynamics[2,5,14] which exhibit complex non-linearity and heterogeneity, i.e. different nodes have different dynamical parameters[14–17] (see Section M.7). These networks function adaptively by adjusting back to a desired steady state despite external perturbations, while they could lose resilience after the removal of keystone nodes which irreversibly pushes the node states into dysfunction that deviates from the desired state[2,5,18,19].

Using our framework, the DRL agent strategically removes nodes based on features of both topology and dynamics until the network collapses, i.e. node states approaching zero, which indicates cell death. We then symbolize the DRL agent, eventually deriving a formula assessing the contribution of individual nodes to network resilience, $d \cdot s$, representing the product of a node's degree and steady state. Here d is the weighted degree, representing the general case that applies to both weighted and unweighted networks. To validate the effectiveness of the discovered metric in different complex networks, we test it on three distinct real-world systems, including gene regulatory networks driven by the Michaelis–Menten dynamics (see Section M.7), neuronal networks[5,15,16] driven by the Wilson–Cowan dynamics[20,21] (see Section M.8), and epidemic networks driven by the Susceptible-Infected-Susceptible (SIS) dynamics[22,23] (see Section M.9).

To validate the discovered $d \cdot s$ formula, we compare it against both existing approaches and corollary topo-dynamic metrics on real-world networks. These variants include using the node state s alone, the product of PageRank and state (PageRank $\cdot s$), and the product of resilience centrality and state (RC $\cdot s$). We evaluated our primary formula and these additional variants across all tested dynamical systems (cellular, neuronal, and epidemic). As shown in Table 1, because resilience is defined based on the dynamics imposed on each network, the topo-dynamic formulas, as expected, dramatically outperform purely topological methods. Among all topo-dynamic metrics, $d \cdot s$ achieves the best or comparable performance, demonstrating the effectiveness of the degree-state coupling discovered by our framework. In contrast, existing analytical approaches (DC[3], RC[8], CI[7], GND[24], EI[13], and CoreHD[25]) and machine learning models (GDM[9] and FINDER[4]) are designed to capture purely topological properties without incorporating network dynamics, thus requiring 56.0% more node removal on average—with a maximum of 74.1% and 73.3%—for gene regulatory networks and epidemic networks, and 19.9% on average for the neuronal networks. Full results are provided in Supplementary Note 1-2, Supplementary Table 1-2, and Supplementary Figure 1-3, where the discovered formula constantly offers the most accurate identification of keystone nodes with up to 77.8% relative improvements.

To comprehensively evaluate the discovered resilience metric across different types of network structures, we leverage well-established network models[17], including erdős-rényi (ER) networks, Barabási–Albert (BA) networks, random partition (RP) networks with communities, and Watts–Strogatz small-world (SW) networks (see details of synthetic network generation in Section M.10). It is important to clarify these synthetic network models may not be representative of all possible real systems, but they are used in this context specifically to test the effectiveness of our dynamics-aware metric under varied topological constraints. Across these network models, the $d \cdot s$ formula consistently provides precise identification of keystone nodes. It substantially reduces the number of removed nodes by over 43.3% for cellular networks, and achieves an even greater improvements of over 56.2% for neuronal networks (see Figure 2a-d, full results in Supplementary Note 3-4 and Supplementary Figure 4-6). Notably, the steady-state value s alone also substantially outperforms purely topological metrics, confirming the primary role of dynamical state information. The degree term d provides additional discriminative power when nodes have similar states but different topological positions — that is, when d and s are not highly correlated. As shown in Figure 2, $d \cdot s$ consistently outperforms s across all synthetic network types, verifying the contribution of the structural term.

Moreover, we compare the identified keystone nodes with ground-truth values obtained by exhaustive search, illustrating

the error relative to the ground-truth value for different methods in Figure 2e-f (the values of the adjacency matrices are scaled down by a weighting factor to reduce the number of removed nodes which allows for an exhaustive search for ground-truth). Specifically, existing methods (DC[3], RC[8], and FINDER[4]) exhibit significant errors, particularly as network size increases, with the largest overestimation of keystone nodes by 7.5%. In contrast, the $d \cdot s$ formula constantly achieves low errors in removal cost regardless of network size, highlighting the accuracy of the discovered resilience metric.

The $d \cdot s$ formula reveals that network resilience is characterized by two macroscopic characteristics, degree d and state s, which capture a node's influence on system resilience from the perspective of network topology and dynamics, respectively. A higher d implies a broader scope of influence, suggesting that the node is more densely connected within the network, while a higher s signifies a stronger and more lasting impact the node has on the entire system. The discovered $d \cdot s$ formula is a resilience metric that accounts for the coupling of network topology and dynamics, addressing a limitation of existing approaches and expanding our understanding of network resilience.

Towards network-level resilience via $d \cdot s$

In practical applications, a systematic metric that captures resilience at the entire network level is still unknown yet highly desired, as it addresses a critical question: which networks are more resilient? This prompts us to extend the discovered node-level $d \cdot s$ formula to a network-level metric. Essentially, $d \cdot s$ indicates that hub nodes with high d may not be dominant in maintaining a network's functionality, unless they also exhibit high states s. Therefore, the correlation between degree and state distributions significantly influences network resilience. Towards this end, we investigate network-level resilience—measured by the number of nodes removed for collapse—under varying correlation of degree and state distributions, using the widely studied scale-free BA networks and homogeneous ER networks.

As predicted in Figure 3a-b, networks are least resilient when degree and state distributions are positively correlated (measured by high Pearson correlation between the distribution of node degree d and node state s, i.e., $\mathrm{corr}(d,s)$), where hub nodes become dominant players again, rendering the system fragile once these hubs are removed. Surprisingly, as the correlation decreases (lower $\mathrm{corr}(d,s)$), networks become substantially more resilient. For instance in Figure 3a-b, the number of removed nodes increases by over 183% and 300% when decorrelating degree and state distributions. The surprising relation between degree-state correlation and network resilience can be explained by the distribution of $d \cdot s$. Specifically, under conditions where degree d is decorrelated with s, the risk of network collapse, indicated by the value of $d \cdot s$, is not concentrated on hub nodes but rather distributed more uniformly across a wide range of nodes, thereby requiring more nodes to be removed to induce the loss of resilience (Figure 3c-d). This predictability of network resilience from its degree-state correlation is also consistently observed across different types of networks with heterogeneous dynamics, where networks of high $\mathrm{corr}(d,s)$ exhibit worse resilience measured by the number of nodes removed for collapse (Figure 4a-d). While classical resilience metrics[2,3] address resilience using only topological characteristics, we find that the correlation between degree and state distributions—which considers both topology and dynamics—can effectively predict network-level resilience, offering an additional perspective for understanding real-world complex systems.

To further examine the derived network-level resilience proxy on real-world systems, we study supply chain networks and ecology networks, where the state s represents throughput of companies and species abundance, respectively. We observe that ecology networks generally display lower correlation between degree and state distributions compared to supply chain networks, with the average difference in $\mathrm{corr}(d,s)$ exceeding 55.9% (see Figure 3e). In supply chain networks, the degree-state correlation decreases with network size, with a substantial drop in $\mathrm{corr}(d,s)$ by 38.8% (see Figure 3f). These patterns are consistent with findings in the literature that natural ecosystems tend to be more resilient than artificial systems[10], and that incorporating diverse additional suppliers enhances supply chain resilience[26]. We note, however, that these observations are descriptive: for supply chain networks, no validated dynamical model is available, and thus a direct resilience analysis as performed for synthetic networks is not feasible. Meanwhile, we also tested the resilience of the same set of ecological networks using a widely adopted mutualistic dynamics[2,5,22]. For each network, we construct both high and low degree-state correlation variants by adjusting the dynamical parameters proportionally and disproportionally to the degree value. We dismantle each network using three independent attack strategies, ranking nodes by d, by s, and by d·s, and define the network's resilience as the minimum number of node removals required to induce collapse across all three strategies. Under this metric-independent definition, networks with low $\mathrm{corr}(d,s)$ consistently require more node removals to collapse than their high-correlation counterparts. For instance, across three ecological networks of 88, 125, and 185 nodes, the high-correlation variants collapse after only 8, 8, and 18 removals, while the low-correlation variants require 12, 35, and 23 removals, respectively (see Figure 4e-f). These findings from real-world supply chain and ecology networks further confirm that the correlation between degree and state distribution ($\mathrm{corr}(d,s)$) indeed serves as a robust and practical indicator for evaluating network-level resilience.

It is worthwhile to note that the $d \cdot s$ formula is obtained from synthetic networks covering a diverse range of $\mathrm{corr}(d,s)$ values to simulate the distribution of real-world networks summarized in Supplementary Figure 14. Meanwhile, the distributions of $\mathrm{corr}(d,s)$ values for tested networks cover even more diverse ranges, where our $d \cdot s$ formula consistently delivers substantial

advantages over existing methods as introduced in the main results, confirming the effectiveness and robustness of our approach.

More results on applying $d \cdot s$ to enhance network resilience are provided in Supplementary Note 8 and Supplementary Figure 9-10, demonstrating that protecting just 3 nodes with the largest $d \cdot s$ values can increase network resilience by over 224%. These findings underscore the practical value of the discovered metric, offering actionable insights to safeguard real-world systems from detrimental disruptions.

Use case I of symbolized reinforcement learning: network robustness

As mentioned before, existing network resilience metrics largely focus on the topological structure. For this distinct problem of network robustness, network functionality sustains as long as the topology remains connected, such as Internet and WWW. In this case, robustness is usually measured by the size of the greatest connected component (GCC)[3,6,7], where a network fully loses its robustness until the GCC size approaches 0 indicating topology disintegration[4,27]. Albert, Jeong and Barabási[3] first investigated network robustness in 2000, where they attacked networks according to node degree (DC) that prioritizes hubs. Since then, research on robustness has been emerging, evidenced by well-established formulas[7,13], computational heuristics[6,24,25], and black-box models[4,6,9,24,28] for identifying keystone nodes of these networks.

We train our DRL agent using widely adopted BA networks, whose scale-free properties mirror the degree distribution found in real-world systems[29]. To ensure the agent could discover the most effective strategy, we provided it with a diverse set of topological node features, including degree, k-core value, clustering coefficient, PageRank value, and betweenness centrality. At each step, the agent receives a reward of the accumulated normalized connectivity[4,27], encouraged to disconnect the network with minimal node removal. We then symbolize the DRL agent. The distillation phase identified the four most important features as degree, average neighbor degree, k-core, and average neighbor k-core. Despite given a much richer feature space, the symbolic regression phase discovered a concise formula $\frac{d^2}{\bar{d}}$, where d and $\bar{d}$ represent the degree and the average neighbor degree, respectively.

We evaluate this formula on an extensive set of 11 real-world networks drawn from a widely used benchmark repository[6]. To ensure a fair and direct comparison of the core node-ranking heuristics, the reinsertion phase[9]–a crucial component for many state-of-the-art algorithms–was omitted for all methods in our main analysis (results with reinsertion are provided in Supplementary Table 6). As illustrated in Table 2, the discovered $\frac{d^2}{\bar{d}}$ formula accurately identify keystone nodes for network robustness, outperforming CI[7] by 1.72% on average, with a maximum of 3.53% improvement, in terms of removal cost measured by accumulative network connectivity (ANC)[4,27]. However, our formula does not achieve the best performance on two networks: on Food-Maayan, CI attains a lower ANC (0.1945 vs. 0.1993), and on Power-Eris, DomiRank32 achieves a lower ANC (0.0509 vs. 0.0736). In both cases, our formula remains the second-best method and outperforms all other baselines. These results reflect a realistic finding: no single metric can dominate across all network types, as each approach may be particularly well-suited to networks with certain structural properties. The formula discovered by our framework depends on the training network topology, and networks whose properties deviate substantially from the training distribution may favor alternative metrics.

While classical metrics such as CI, denoted by $(d_i - 1)\sum_{j \in \partial \,\mathrm{Ball}(i,l)} (d_j - 1)$, provide a method for identifying influential nodes, their interpretation can be complex and dependent on parameters like the ball radius l. In contrast, our discovered metric offers an elegant and parameter-free formula, $\frac{d^2}{\bar{d}}$, which provides a more direct intuition. Rewriting $\frac{d^2}{\bar{d}}$ as $\frac{d}{\bar{d}} \cdot d$, the discovered formula suggests that a node's influence on network robustness is determined by not only the degree of itself (d), but also its competitiveness over its local environment or neighbors ($\frac{d}{\bar{d}}$). It predicts that hub nodes with both large d and $\bar{d}$ are less critical due to their interchangeability with neighbors, reflected by their lower $\frac{d}{\bar{d}}$. Meanwhile, nodes with high degree connecting to less central neighbors (large d, small $\bar{d}$) are exceptionally critical, as they often serve as crucial bridges between weakly linked components. This aligns with findings that bridges of bridges can be preferential targets[9]. In this way, the discovery of $\frac{d^2}{\bar{d}}$ not only refines well-established metrics with higher measured accuracy but also offers a more concise and physically interpretable model of robustness, indicating that machine learning can advance classical network metrics.

The formula's effectiveness can be further understood by its strong ability to identify a network's articulation points (or cut vertices)—nodes whose removal increases the number of connected components. The mathematical signature of a high degree (d) combined with a low average neighbor degree ($\hat{d}$) is a clear indicator of such nodes, as they often serve as critical bridges connecting otherwise disparate parts of the network. To empirically validate this connection, we conducted an analysis across our real-world test networks. After ranking all nodes by their $\frac{d^2}{\bar{d}}$ score, we measured the fraction of true articulation points captured within the top-ranked nodes. The results (Supplementary Table 7) are compelling: The formula consistently prioritizes these nodes with high precision across all datasets, identifying an average of 87.1% true articulation points in the top-10 ranked nodes and 83.6% in the top-20. For several networks, this fraction reaches 100%. This provides strong empirical evidence that the success of our discovered formula is rooted in its ability to efficiently and accurately pinpoint the most structurally critical

nodes in a network.

It is worthwhile to note that the central contribution of our paper lies in discovering metrics for dynamical resilience. The application of our framework to topological robustness, while yielding compelling results, is primarily intended to showcase its versatility in a distinct domain. We acknowledge the profound complexity of the network dismantling task. Accordingly, the simple formula $\frac{d^2}{\bar{d}}$ is not proposed as a definitive, state-of-the-art solution, but rather as an insightful and interpretable heuristic that advances our understanding of the factors governing network robustness.

To examine whether the discovered formula depends on the training topology, we further apply our self-inductive framework to SW networks. The GNNExplainer identifies the four most important features as degree, average neighbor k-core, betweenness centrality, and average neighbor degree — notably different from those identified on BA networks (degree, average neighbor degree, k-core, average neighbor k-core). The subsequent symbolic regression discovers a topological robustness formula $d^2-\bar{d}$ for SW networks, in contrast to $d^2/\bar{d}$ for BA networks. These results confirm that the choice of training topology influences the discovered formula, as networks with different structural properties emphasize different topological features. Importantly, we interpret this result as evidence for the flexibility of our self-inductive framework rather than evidence that different network classes should yield the same or similar formulas. Different topologies naturally emphasize different structural features, and applying the same methodology to other classes, including regular, lattice-like, modular, or otherwise complex networks, may produce distinct expressions.

Use case II of symbolized reinforcement learning: improving over resilience centrality

Gao, Barzel, and Barabási[2] studied the functional resilience of complex networks by introducing linear and homogeneous assumptions on networks dynamics, i.e. all nodes share the same dynamical parameters. Through dimension reduction analysis, they obtained a universal β formula quantifying network resilience, based on which resilience centrality (RC[8]: $2\bar{d}+d(d-2\beta)$), was derived to identify keystone nodes.

In this scenario featuring homogeneous system dynamics, we train our DRL agent on gene regulatory dynamics, and evaluate on both real-world gene regulatory and neuronal networks[2,5,14–16]. After symbolizing the agent to decipher its node selection strategy, we derive a resilience formula, $2\bar{d}+d(\bar{d}-2\beta)$. To ensure a fair and comprehensive evaluation, we benchmarked this discovered formula not only against the original Resilience Centrality (RC) but also against a suite of strong topological baselines, including DC[3], CI[7], PageRank[30], and DomiRank[28]. The results of this expanded benchmark are presented in Table 3. Our discovered formula demonstrates a clear advantage. It not only improves upon the original RC metric in all 8 test cases but also outperforms the other strong baselines in 7 out of 8 cases. These results, with an average improvement of 22.5% and 11.9% over RC for two types of gene regulatory networks and a notable maximum improvement of 37.5%, confirm that our framework can robustly refine existing formulas even under more challenging and equitable comparisons.

Building upon the well-established β formula[2] and RC formula[8], our discovered $2\bar{d}+d(\bar{d}-2\beta)$ formula suggests that the ability of a node to affect network resilience relates to both its degree and nearest-neighbor degree. Notably, this formula improves upon RC by modifying one term, replacing $d-2\beta$ with $\bar{d}-2\beta$. This change suggests that a node's degree has a positive impact on its importance to network resilience when $\bar{d}$ exceeds 2β, rather than d surpassing β, as previously suggested. In fact, RC introduces approximations during its derivation, which can lead to non-neglectable inaccuracies. The performance improvements indicate that, in addition to deriving additional formulas, our framework can make measurable improvements to existing ones, extending our understanding of network resilience.

Discussion

In this work, we present a symbolized reinforcement learning framework which moves beyond black-box prediction to discover interpretable mathematical formulas for complex network problems. Featuring a self-inductive manner, the key difference between our approach and prevailing machine learning methods[4,9,28,31] lies in the non-trivial transformation from a black-box DRL agent to insightful and human-understandable mathematical formulas. Notably, our approach is capable of refining classical network metrics traditionally derived through extensive derivations, such as the complicated dimension reduction analysis used in the universal β formula[2] (Supplementary Note 5, Supplementary Table 4-5 and Supplementary Figure 7). Our results demonstrate that symbolized reinforcement learning holds significant untapped potential for pushing the boundary of complexity science.

Using our framework, we have successfully established a comprehensive set of centrality metrics that expand our understanding of network resilience. On the one hand, our framework refines existing formulas in analyzable systems, which has attracted substantial research efforts in the past 10 years, including both topological robustness formulas and functional resilience formulas. On the other hand, our framework discovers resilience metrics in systems that were believed to be intractable using current analytical tools, leading to a resilience formula accounting for the coupling of network topology and heterogeneous dynamics. The discovered formulas represent measurable improvements in network robustness and resilience with potential

applications to real-world complex systems, including the early-warning application detailed in Supplementary Note 7 and information relevant to designing practical systems resilient against perturbations.

A crucial aspect of our framework is the role of the GNN. Since the final symbolic formulas are expressed using node features, one might question if a simpler MLP could replace the GNN without loss of performance. To address this, we performed an ablation study on the topological robustness task, training an alternative agent with an MLP encoder. The results confirmed the necessity of the GNN: the GNN-based agent consistently outperformed the MLP-based agent in identifying keystone nodes (e.g., achieving lower removal costs of 0.4792 vs. 0.4938 on the Bay-Dry network and 0.4803 vs. 0.4915 on the Bay-Wet network). Furthermore, symbolic regression on the MLP's strategy only rediscovered known heuristics like degree (d), while the GNN's topology-aware strategy was required to derive the $\frac{d^2}{\hat{d}}$ formula. Therefore, the discovered formulas should not be seen as a direct explanation of the GNN's internal mechanism which can be far more complicated, but rather as human-interpretable symbolic approximations that effectively capture the GNN's strategy. In other words, the GNN is essential for finding a high-quality solution in the first place; our symbolization pipeline then makes that solution accessible and understandable.

The benefits of our proposed symbolized reinforcement learning framework are multi-fold. It can help achieve improved generalization towards unseen data, as evidenced by Supplementary Note 6 and Supplementary Figure 8a where the symbolized $d \cdot s$ formula outperforms its DRL agent precursor on networks that are not included in the training data. Additionally, it offers super-fast inference speed, which sidesteps the need for complete computations of neural networks. For example in Supplementary Note 6 and Supplementary Figure 8b, the $d \cdot s$ formula reduces inference time by over 34.5%, with speedup exceeding 50.1% for large networks. This fusion of generalizability and efficiency highlights the potential of symbolized reinforcement learning for both formulaic understanding and practical applications in complex networks.

We note that a current limitation of the network resilience and dismantling field is the absence of a standardized benchmarking framework[6], while existing studies often rely on different sets of benchmark networks, dynamical models, and evaluation protocols. This heterogeneity makes it difficult to draw definitive conclusions about the relative performance of competing methods across studies. Though we have endeavored to conduct fair comparisons by adopting established benchmark repositories and including a diverse set of network types and dynamics, our benchmark selection, like those in prior works[4,6,28], is inevitably incomplete. An important direction for future work is the construction of a standardized benchmark suite for network resilience and dismantling tasks, encompassing diverse topologies, dynamical models, and evaluation metrics, which would facilitate more transparent comparisons and accelerate methodological progress.

A direction for future research is to explore the application of our approach beyond network resilience, such as discovering formulas on network growth[32,33] and percolation[34], which is conceptually the inverse of network dismantling. In particular, our framework can be adapted to study percolation by inverting the RL agent's objective as node addition for maximizing connectivity instead of removing nodes to break the network. The symbolization phase would then distill the agent's strategies into formulas for identifying optimal spreader nodes or influence centrality. This would extend our approach to discover additional formulas of influence maximization and efficient network assembly. Moreover, our self-inductive framework has far-reaching implications in the AI for science field. Besides complexity science, the symbolized reinforcement learning approach can also be extended to inspire innovative discoveries in other domains where formulaic understanding has traditionally been achieved by human experts, such as mathematics[35,36], biology[37,38] and materials[39]. More broadly, our work could serve as a prototype for machine learning systems that operate in a human-understandable way, assisting in the acquisition of scientific insights.

Methods

M.1 Overview

In this study, we introduce a machine learning framework designed to deepen human understanding in complex networks, particularly in high-dimensional systems characterized by non-linear and heterogeneous dynamics—domains where traditional approaches relying on analytical derivations often fall short. The proposed framework is not limited to a specific task but serves as a versatile tool for scientific discovery in complex networks. Specifically, we leverage this framework to quantify network resilience and identify associated critical nodes. As shown in Figure 1, our framework employs machine learning to accomplish the complicated network dismantling task, searching critical nodes with RL, then symbolize itself into network resilience metrics. In this self-inductive framework, human experts are involved to define the problem and refine the candidate results generated by the machine learning model, while leaving the complicated and tedious searching procedure to the machine learning model[12]. We now provide a brief overview of resilience metric discovery using our framework (Supplementary Figure 13).

- **Define.** In order to employ machine learning models, the initial step involves framing the task as a well-defined computational problem with appropriate input and output specifications. Since the ground-truth of keystone nodes can only be obtained by

exhaustive search, a practically infeasible endeavor, we approach the task by approximating it as an optimization problem, i.e. the network dismantling problem. In specific, we remove nodes from the network to compromise its resilience, with the objective of minimizing the total cost of node removal. The criteria for selecting nodes thus serve as the indicator of node importance.

- **Search.** The formulated network dismantling problem exhibits an enormous solution space, which requires efficient search to obtain a decent strategy. Towards this end, we frame the problem as a Markov decision process (MDP), where one node is removed per step until the network loses resilience. We design a deep reinforcement learning (RL) agent to solve this MDP, with the object of concluding the episode with the fewest steps. We create an environment to simulate high-dimensional networks characterized by non-linear and heterogeneous dynamics. This environment evaluates the resilience of the current system and provides feedback to the DRL agent. A GNN based state encoder is designed to capture rich topological and dynamical features of nodes, which is shared between a policy network and a value network, with the former selecting nodes for removal and the latter predicting network resilience based on the learned network representations, respectively. Through millions of interactions with the environment, the RL agent eventually attains a superhuman level performance in quantifying network resilience and identifying keystone nodes, resulting in a highly accurate solution to the network dismantling problem that is characterized by neural networks.

- **Distill.** The DRL model operates as a black box, offering limited explainability regarding how it derives predicted node importance scores, which impedes human comprehension of the contribution of different nodes to the overall network resilience. As the node removal decision is made based on node embeddings extracted from rich node features, we distill the DRL model with explainable machine learning techniques to scrutinize feature importance, attributing the model prediction to individual node features. Among all input node features, certain features emerge as pivotal that dominate the agent's decision-making process, while others prove insignificant in influencing model predictions. Through explainable machine learning, we eliminate noise from the complicated DRL model, and distill a streamlined version of the model that retains only the important features.

- **Discover.** Ultimately, we aim to translate the effective strategy of the machine learning model into a human-understandable form, discovering simple mathematical rules that approximate its behavior in identifying critical nodes. In pursuit of this goal, we discover mathematical connections between important features and critical nodes with symbolic regression (SR), to derive a physically insightful index of node importance from the DRL agent. A diverse set of complex networks is employed to construct a dataset for SR, based on which a mathematical formula is discovered, predicting the nodes selected by the DRL model from the important features distilled by explainable machine learning techniques. This resultant human understandable physical metric faithfully replicates the behaviors of the DRL model while in the meantime offering greater interpretability. Through employing SR to correlate the DRL policy with the important node features, we generate information that further supports our understanding of the factors underlying the performance of machine learning models.

- **Refine.** The above discovery process can generate multiple candidate formulas. Meanwhile, the extracted equations may exhibit complexity, harboring redundant terms or unnecessary constant values. Therefore, as the concluding step in our proposed framework, human-in-the-loop collaboration is employed to refine the generated formulaic results. Ultimately, a simplified and concise metric is derived that comprehensively captures the importance of different nodes to the overall network resilience. The resulting metric is designed to be easily understood by human, enhancing its utility in practical applications such as network protection and network design.

With the outlined framework, we take full advantage of the computational power of machine learning models not only to obtain an accurate model but, more importantly, to generate tangible and insightful formulas about network resilience. Diverging from prevailing methods that culminate in black-box models, our approach harnesses machine learning to not only solve the problem but also understand how machine learning achieves the success, leading to the generation of human-understandable metrics. Subsequently, we elaborate on the design of our proposed framework.

M.2 Define as an optimization problem

Complex systems can be described as a network of N components (nodes) $V = \{1,\dots,N\}$ whose activities $\mathbf{x} = (x_1,\dots,x_N)$ are driven by heterogeneous self-dynamics $\mathbf{F} = \{F_1,\dots,F_N\}$ and interactions between them $\mathbf{G} = \{\dots,G_{ij},\dots\}$, as characterized by the coupled equations[15,16]

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = F_i(x_i) + \sum_{j=1}^{N} A_{ij} G_{ij}(x_i,x_j), \tag{1}$$

where non-linear dynamic functions[5,14,17,40] F_i and G_{ij} are intricately intertwined through the interaction matrix $\mathbf{A}$. Despite frequent external perturbations and environmental changes that push $\mathbf{x}$ away from its desired functional state $\mathbf{x}^{\mathrm{H}}$, system failures are rarely observed; instead, these systems often display a surprising degree of resilience, being able to adjust back to $\mathbf{x}^{\mathrm{H}}$ and maintain their basic functionality[2,5,18,19]. However, errors on the components (node loss) can lead to an irreversible phase transition, driving the system into a non-resilient basin manifesting dysfunction and breakdowns upon the removal of a subset of nodes $V_c \subset V$,

$$\text{resilient} \quad \mathcal{R}(\mathbf{A},[\mathbf{F},\mathbf{G}]) = 1 \quad \Leftrightarrow \quad \mathbf{x}(t\to\infty) = \mathbf{x}^{\mathrm{H}}, \ \ \forall \mathbf{x}(t=0) \tag{2}$$

$$\text{non-resilient} \quad \mathcal{R}(\mathbf{A}[V\setminus V_c],[\mathbf{F},\mathbf{G}]) = 0 \quad \Leftrightarrow \quad \mathbf{x}(t\to\infty) \neq \mathbf{x}^{\mathrm{H}}, \ \ \exists \mathbf{x}(t=0) \tag{3}$$

Here $\mathcal{R}$ measures the resilience of the system which cannot be expressed as a mathematically analytic equation due to the non-linearity and heterogeneity of $[\mathbf{F},\mathbf{G}]$. While all nodes influence the functionality of the entire system, the contributions of different nodes to the overall network resilience are not equal due to the prevalent asymmetry property in real-world complex networks. Notably, a small subset of critical nodes $V_c \in V$ can have a significant impact on network resilience, and their removal can drastically degrade the network's basic functionality; see Figure 1a (bottom-left). The discovery of critical nodes in network resilience can be formulated as an optimization problem as follows,

$$\underset{V_c}{\textbf{minimize}} \quad \|V_c\|, \tag{4}$$

$$\textbf{subject to} \quad \mathcal{R}(\mathbf{A}[V\setminus V_c],[\mathbf{F},\mathbf{G}]) = 0, \tag{5}$$

$$\textbf{where} \quad V_c \subset V \tag{6}$$

Here, the goal is to minimize the number of removed nodes $\|V_c\|$ to compromise the network's resilience. The nodes selected for removal represent the discovered critical nodes V_c, and the criteria for selection serves as a useful index for node importance. The problem (4-6) in its original form requires generating all critical nodes at once which is intractable, thus it is usually transformed to a recursive version as follows,

$$\underset{\mathcal{Q}}{\textbf{minimize}} \quad \kappa, \tag{7}$$

$$\textbf{subject to} \quad \mathcal{R}(\mathbf{A}[V\setminus V_c^{\kappa}],[\mathbf{F},\mathbf{G}]) = 0, \tag{8}$$

$$\textbf{where} \quad V_c^0 = \varnothing, \tag{9}$$

$$V_c^{\kappa} = V_c^{\kappa-1} \cup \{v_c^{\kappa}\}, \tag{10}$$

$$v_c^{\kappa} = \underset{i}{\operatorname{argmax}}\{\mathcal{Q}(i;\mathbf{A}[V\setminus V_c^{\kappa-1}],[\mathbf{F},\mathbf{G}]) \mid i \in V\setminus V_c^{\kappa-1}\}, \ \ \kappa = 1,2,\ldots \tag{11}$$

M.3 Search with DRL

MDP definition. Finding the optimal set of keystone nodes V_c is a computationally challenging problem[11], known to be NP-hard[11], whose ground-truth solutions can only be obtained through exhaustive search, which is infeasible in practical scenarios involving large-scale networks. To address the challenge brought by the large solution space, we leverage DRL to facilitate efficient search, aiming to train an intelligent agent capable of identifying keystone nodes. To achieve this, we first reformulate the problem as a Markov Decision Process (MDP), which consists of the following key elements:

- **State.** This describes the current conditions of the complex network, including the topology structure $\mathbf{A}_t = (V_t, E_t)$, and corresponding comprehensive node features considering both topology and dynamics. Specifically, we characterize each node with an 11-dimensional feature set, which includes attributes such as `degree`, `maximum edge weight`, `neighbor degree`, `resilience centrality`, `dynamical parameters(3)`, `state`, `state derivative`, `neightbor state`, and `neighbor state derivative`.
- **Action.** At each step of the MDP, the DRL agent removes one single node, attempting to fail the network's resilience. Therefore the action a_t indicates the selected node for removal at step t.
- **Reward.** Since the goal is to minimize the number of removed nodes, the reward r_t at each step is set to -1, encouraging the process to conclude promptly with the fewest steps. The process terminates when the network loses its resilience, which is precisely defined by the system-specific resilience functions ($\mathcal{R} = 0$) detailed in Section M.7 (Cellular dynamics) and Section M.8 (Neuronal dynamics). The agent is trained to achieve a higher long-term return, which is the cumulative sum of the rewards at each step. Consequently, it learns to identify the most influential nodes for removal, compromising the network's resilience with fewer steps.

- **Transition.** It describes how the network changes with the current action, where the node a_t is removed along with all edges connected to it. In addition, the removal of the node a_t may make the network disconnected into multiple components. In such cases, we focus on the residual greatest connected component (GCC) subsequently in accordance with existing literature[2]. Therefore, the network $\mathbf{A}_t$ transitions to $\mathbf{A}_{t+1}$, which is either $\mathbf{A}_t \setminus a_t$ or $\mathrm{GCC}(\mathbf{A}_t \setminus a_t)$, depending on the connectivity of the residual network.

Figure 1a (top-left) and b and Supplementary Figure 13b demonstrate our proposed DRL approach, where an agent interacts with an environment, with the object of compromising the network's resilience using the fewest steps of node removal. To begin with, we construct an environment to simulate the complex network. This environment implements the previously described transition process for each action, thereby altering the network structures. It generates the state by simulating the trajectories of the network given the current topology $\mathbf{A}_t$ and dynamic functions $[\mathbf{F}, \mathbf{G}]$ described with ordinary derivative equations (ODE) as equations (1). Meanwhile, it evaluates the resilience of the current network by perturbing the states $\mathbf{x}$ and evaluating whether the system can remain at its desired state $\mathbf{x}^{\mathrm{H}}$ or featuring an undesired state $\mathbf{x}^{\mathrm{L}}$, providing rewards to the agent, and then terminates the episode once the network loses its resilience (see details of resilience calculation in Section M.7 and M.8).

GNN state encoder. We develop an agent consisting of a state encoder for learning node representations, a policy network for selecting nodes for removal, and a value network for estimating network resilience. To capture the intricate topological and dynamical features, we employ Graph Neural Networks (GNN)[41] to encode the rich node features, resulting in dense embeddings of different nodes. Specifically, we conduct message passing and neighbor aggregation over the graph, through stacking several linear transformation layers and non-linear activation layers. The 11-dimensional node features serve as input attributes to the GNN, leading to the generation of initial node embeddings as follows:

$$\mathbf{X}_i = \eta_1(i) \parallel \eta_2(i) \parallel \cdots \parallel \eta_{11}(i), \tag{12}$$

$$\mathbf{h}_i^0 = \tanh(\mathbf{W}^0 \mathbf{X}_i), \tag{13}$$

where $\mathbf{X}_i$ denotes the input attributes for node i, and $\parallel$ signifies the concatenation of the 11 different features as introduced previously. We transform the input node attributes with a linear transformation matrix $\mathbf{W}^0$ and the tanh activation function.

We then propagate the node embeddings over the graph and aggregate messages from neighbors to iteratively update node embeddings. This recursive process involves several graph convolutional layers, as expressed below,

$$\mathbf{h}_i^l = \tanh(\mathbf{W}^l \sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{e_{j,i}}{\sqrt{\hat{d}_j \hat{d}_i}} \mathbf{h}_j^{l-1}), l = 1, 2, \cdots, L, \tag{14}$$

where $\hat{d}_i = 1 + \sum_{j \in \mathcal{N}(i)} e_{j,i}$, and $e_{j,i}$ denotes the edge weight from node j to node i. For each node, this process absorbs information from its neighboring nodes to update its own embedding, utilizing a linear transformation layer $\mathbf{W}^l$ and the tanh activation unit. Through the stacking of multiple graph convolutional layers, we obtain the final node embedding $\mathbf{h}_i^L$, where L is a hyper-parameter in our model. The learned node representations encapsulate both the topological and dynamical features of each node, and meanwhile consider the influence of high-order neighbor nodes. These semantic embeddings inform subsequent action making and return prediction, with the GNN stated encoder shared between policy and value networks.

Policy and value network. We employ the actor-critic approach, with a policy network taking actions and a value network estimating returns at each step. Specifically, the policy network assigns scores to each node, based on which one single node is selected for removal. Node scores are computed from the node embeddings, extracted by the GNN state encoder. On the other hand, the value network takes graph-level representations and predicts the return, which captures the resilience conditions of the current complex system. To achieve a comprehensive graph-level representation, summarizing the information of the entire network, we use the average of all node embeddings. For both the policy and value networks, we employ multi-layer perceptrons (MLP) models, expressed as follows,

$$s_i = \mathrm{MLP_p}(\mathbf{h}_i^L), \tag{15}$$

$$\hat{r}_t = \mathrm{MLP_v}(\frac{1}{|V|} \sum_{i \in V} \mathbf{h}_i^L), \tag{16}$$

where s_i is the predicted score for node i, $\hat{r}_t$ is the estimated return value, and $\mathrm{MLP_p}$ and $\mathrm{MLP_v}$ consist of several consecutive linear feed-forward layers and non-linear activation layers. During training, nodes are selected by sampling based on their corresponding scores, with the selection probability proportional to their scores as follow,

$$\mathrm{Prob}(i) = \frac{\mathrm{e}^{s_i}}{\sum_{j \in V} \mathrm{e}^{s_j}}, \tag{17}$$

where we utilize this softmax function to construct a proper probability distribution. When applying a well-trained model to identify keystone nodes, we select nodes greedily at each step as follow,

$$a_i = \text{argmax}\{s_i \mid i \in V\}, \tag{18}$$

where the node with the largest predicted score is selected. It is noteworthy that the score s_i reflects the importance of node i, and the estimated value $\hat{r}_t$ indicates how resilient the current network is. In essence, the policy network, along with the GNN state encoder, functions as a black-box estimator, assessing the contribution of different nodes to the overall network resilience. Similarly, the value network, coupled with the GNN state encoder, serves as an estimator evaluating the resilience of a complex network. However, the parameters of the neural networks are challenging to interpret directly by humans. We will later open up these black-box estimators, unveiling tangible and insightful formulas that are much more perceivable to humans.

Model training. We implement both the environment and the agent with the Stable-Baselines3 framework[42], and train the agent using the Proximal Policy Optimization (PPO) algorithm[43]. Specifically, we utilize the following surrogate clipped object as the policy loss, encouraging the policy to make adaptive updates,

$$r_t(\theta) = \frac{\pi_{\theta_{\text{new}}}(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}, \tag{19}$$

$$\hat{A}_t = Q(s_t, a_t) - v(s_t), \tag{20}$$

$$L_{\text{p}}(\theta) = \min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t), \tag{21}$$

where θ denotes the model parameters, $r_t(\theta)$ is the ratio of the probability under the updated policy to the old policy, $\hat{A}_t$ is the estimate advantage at time t, and ε is the clip hyper-parameter to prevent large model updates. With the clipping mechanism, we stabilize the training process while guarantee safe exploration. After multiple iterations of policy optimization, the policy gradually approaches optimal solutions with each iteration improving a small step near the original policy.

Due to the enormous solution space, it is impractical for the agent to enumerate all feasible node removal strategies during interactions with the environment. To address this challenge, we include an entropy loss on the action sampling probability distribution to encourage the policy to be more stochastic as follow,

$$L_{\text{e}} = -\text{Entropy}(\{\text{Prob}(i) \mid i \in V\}). \tag{22}$$

Here entropy encourages the agent to explore a broader range of actions, even those with lower probabilities. This proves particularly beneficial in the early stages of learning or when facing environments with high uncertainty, where increased exploration aids the agent in discovering better policies and improving its overall performance. In addition, it helps prevent the policy from becoming excessively deterministic and overfitting to the observed data. By penalizing policies that assign very high probabilities to a single action, entropy loss promotes a more balanced distribution over actions.

We adopt mean squared error between the estimated return and the ground-truth to supervise the value network as follow,

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \tag{23}$$

$$L_{\text{v}} = \text{MSE}(\hat{r}_t, R_t), \tag{24}$$

where R_t is the long-term return at time step t, a cumulative discounted aggregation of per-step reward, and $\hat{r}_t$ is the predicted return by the value network according to equation (16). The agent is optimized jointly by the three loss functions as follow,

$$L = L_{\text{p}} + \lambda_1 L_{\text{e}} + \lambda_2 L_{\text{v}}, \tag{25}$$

where λ_1 and λ_2 are hyper-parameters in our model controlling the weight of different loss functions.

We train the DRL model with cellular networks, running the Michaelis–Menten dynamic function on 30 different scale-free topologies, each consisting of 80 nodes. The choice of 30 networks is an empirical balance between ensuring sample diversity and maintaining a tolerable training cost. These networks are used exclusively for training, while all validation is performed on unseen synthetic and real-world networks. Training networks are generated via a configuration model approach: first, a degree sequence is drawn from a power-law distribution ($\gamma = 2.0$) and scaled to have a predefined average degree ($\bar{d} = 6$). After rounding the degrees to the nearest integers, we construct a pseudograph by randomly connecting nodes to match the degree sequence. Finally, any parallel edges or self-loops are removed to produce the final simple, undirected training graphs. During training episodes, the environment iterates over the 30 distinct graphs, with the agent interacting by removing one node from the current topology at each step. The DRL model is trained for 1 million steps, allowing the agent to observe

thousands of induced residual topologies after the removal of identified nodes. Meanwhile, we periodically evaluate the average performance of identifying keystone nodes across the 30 networks to monitor the model training process. After training, we select the checkpoint that achieves the best performance (see Supplementary Note 4.1 and Figure 6a for convergence details). The resulting DRL model is proficient in searching for keystone nodes of network resilience with higher accuracy, yet with limited explainability regarding why it selects specific nodes. Such explainability is crucial for comprehending the success of machine learning models, ultimately fostering insights and enhancing human understanding of network resilience.

Hyper-parameters. In our experiments, we set the number of GNN layers L as 2, dimensions of node embeddings $\mathbf{W}^0$ as 32, number of hidden layers in policy and value networks MLP_p and MLP_v as 2 with hidden dimensions as 32. We use tanh as activation functions. We use a learning rate of 0.0004 and a batch size of 500 for RL training. The loss weights for entropy λ_1 and value λ_2 is 0.01 and 0.5, respectively.

M.4 Distill with explainable machine learning

Binary classifier GNN for distillation. The DRL agent is able to accurately identify critical nodes V_c of network resilience, yet it is a black-box model lacking transparency in how it selects these nodes. Particularly, the node selection decision is made based on learned node embeddings by a complicated GNN model, which involves multiple layers of message propagation and non-linear activation units. To understand the GNN based policy, we utilize explainable machine learning to distill key information from the black-box model. In specific, we first construct a separate GNN model Θ by combining the GNN state encoder and the policy network in equations (12-15), as expressed below,

$$\hat{y}_i = \text{MLP}_\text{p}(\tanh(\mathbf{W}^L \cdots \tanh(\mathbf{W}^2 \sum_{j\in\mathcal{N}(i)\cup\{i\}} \frac{e_{i,i}}{\sqrt{\hat{d}_j\hat{d}_i}} \tanh(\mathbf{W}^1 \sum_{j\in\mathcal{N}(i)\cup\{i\}} \frac{e_{i,i}}{\sqrt{\hat{d}_j\hat{d}_i}} \tanh(\mathbf{W}^0[\eta_1(i) \parallel \eta_2(i) \parallel \cdots \parallel \eta_{11}(i)])))))). \tag{26}$$

Here this separate GNN accomplishes a node classification task, predicting whether each node is a critical node in V_c, and $\hat{y}_i$ is the raw predicted score for node i. Stripping the value network and all RL training components, we reserve the major functional module, a GNN based binary classifier, denoted as Θ, that makes predictions for each node. We set the parameters of this separate GNN to the values of the corresponding layers in the RL agent, which guarantees that it can fully reproduce the results of the RL agent.

Mutual information maximization. With this separate GNN binary classifier Θ, we explain how it makes the predictions using a widely acknowledged explainable machine learning tool, GNNExplainer[44]. In Figure 1a (top-right) and c and Supplementary Figure 13c we illustrate the process of distilling the complicated model Θ. Specifically, we attribute the prediction of GNN to the input node features, investigating the importance of different features. As introduced previously, we incorporate 11 features regarding network topology and dynamics, among which are important features that are influential to the model prediction and insignificant features that contribute little information to the model. Therefore, we aim to find a subset of important node features $\Omega = \{\omega_1, \ldots, \omega_P\} \subset \{\eta_1, \ldots, \eta_{11}\}$, such that removing them leads to drastic deterioration of model prediction. From the perspective of mutual information, it can be expressed as follow,

$$\underset{\Omega}{\textbf{maximize}} \quad \text{MI}(Y, \Omega) = H(Y) - H(Y \mid X = \Omega), \tag{27}$$

where MI calculates the mutual information between the model prediction Y and the important features Ω, and H calculates the entropy. As the entropy $H(Y)$ for the prediction of the fixed well-trained model Θ is a constant, we are minimizing the conditional entropy as follow,

$$H(Y \mid X = \Omega) = -\mathbb{E}_{Y|X=\Omega}[\log_{P_\Theta}(Y \mid X = \Omega)]. \tag{28}$$

Meanwhile, we would like the identified important features Ω to be a compact set, containing as few features as possible.

Learning feature selection mask. To obtain important features Ω, we learn a feature selection mask $\mathbf{M}_\text{s} \in \{0,1\}^{11}$ for each node, through which we calculate the masked feature as $\hat{\mathbf{X}}_i = \mathbf{X}_i \otimes \mathbf{M}_\text{s}$. The binary mask is non-differentiable for back-propagation, thus we smooth it into real values using the sigmoid function: $\mathbf{M}_\text{s} = \text{sigmoid}(\mathbf{M})$ where $\mathbf{M} \in \mathbb{R}^{11}$ and is initialized with an empirical distribution $\mathcal{N}(0, 0.1)$. With the complete model Θ and the full input features $\mathbf{X}_i$, we can obtain the original model prediction y; while using the masked input features $\hat{\mathbf{X}}_i$, the model predicts $\hat{y}$ that tends to exhibit errors in

comparison to y. Therefore, to identify important features, we optimize $\mathbf{M}$ with the following loss functions,

$$L_{\mathrm{MI}} = -\sum_{\mathbf{A}}\sum_{i=1}^{N_{\mathbf{A}}} y_i \log\sigma(\hat{y}_i) + (1-y_i)\log(1-\sigma(\hat{y}_i)), \tag{29}$$

$$L_{\mathrm{S}} = -\sum_{\mathbf{A}}\sum_{i=1}^{N_{\mathbf{A}}}\sum_{j=1}^{11} M_{ij}\log M_{ij} + (1-M_{ij})\log(1-M_{ij}), \tag{30}$$

$$L = L_{\mathrm{MI}} + \lambda_3 L_{\mathrm{S}}, \tag{31}$$

where we calculate the loss for all $N_{\mathbf{A}}$ nodes of each training network $\mathbf{A}$. The first term L_{MI} captures whether the masked features can reproduce the model prediction, representing mutual information between model prediction and the selected features. Meanwhile, the second term L_{S} computes the entropy of the feature selection mask, penalizing the inclusion of a large set of important features and encouraging a focus on a sparse subset of features. We introduce a hyper-parameter λ_3 to combine these two terms, optimizing the feature selection mask $\mathbf{M}$ to achieve high mutual information between the selected features and the model prediction, while simultaneously minimizing the number of included features.

Identifying important features. We train the feature selection mask $\mathbf{M}$ with the adopted 30 networks $\mathbf{A}$ of 80 nodes for the DRL agent, and finally we obtain a complete mask $\mathbf{M} \in \mathbb{R}^{(30\times 80)\times 11}$ indicating the importance of different features in various networks. Specifically, if a particular feature η_k is important, the corresponding weights in $\mathbf{W}^0$ tend to be large, and masking this feature will lead to drastic performance deterioration. Therefore, by optimizing the mask $\mathbf{M}$ with equation (31), the corresponding multiplication mask score M_{ijk} will be large. Then we compute the overall feature importance scores by aggregating the results of all 30 networks,

$$S_k = \sum_{i=1}^{30}\sum_{j=1}^{80} M_{ijk}, \quad i = 1,\ldots,11 \tag{32}$$

The final important features Ω are identified according to the computed importance scores,

$$\Omega = \{\omega_1,\ldots,\omega_P\} = \mathrm{topP}\{S_1,\ldots,S_{11}\}. \tag{33}$$

With Ω, we distill the intricate GNN model Θ into a streamlined version, eliminating the noise of it and reserving only the important features that dominate its prediction (see Supplementary Note 4.2 and Figure 6b).

M.5 Discover with SR

Symbolic regression mapping Ω to V_c. To approximate the DRL model in a human-understanable way, we discover the hidden mathematical connections between important features Ω and the identified critical nodes V_c, establishing formulaic mapping with symbolic regression (SR) from Ω to $\Theta(\mathbf{X})$. In other words, we derive a physically insightful index θ from Θ to measure the contribution of different nodes to the network resilience, replacing the complicated GNN model Θ with a human-understandable formula θ that achieves comparable or even better performance in solving the problem (7-11) while providing greater interpretability and supporting downstream applications such as system enhancement and protection. In Figure 1a (top-right) and d and Supplementary Figure 13d we illustrate the equation discovery process given the previously obtained Θ and Ω. Specifically, the equation discovery process contains the following key elements,

- **Data.** We use the 30 synthetic networks for training the DRL model as the raw data for SR, where the DRL model achieves the best performance in comparison to all existing baselines. Specifically, we construct an SR dataset, containing the important node features identified by GNNExplainer and labels indicating whether each node is selected by the DRL model.
- **Primitives.** They are the base functions for SR, serving as the basic ingredients for constructing mathematical equations that combines these primitives. As we have distilled the important features that play pivotal roles in the model prediction—the DRL model also combines these features to measure node importance albeit in a complicated way using neural networks—we adopt these identified important features Ω as primitives.
- **Target.** We would like the derived formula to mimic the behavior of the complicated GNN model Θ, thus we utilize the output logit values for each node as the target for SR. Specifically, we use the $\hat{y}_i$ in equation (26) of each node as the target to predict from the node's corresponding important features (primitives) Ω.
- **Operators:** These are the mathematical operations that combine various elements, whether primitives or sub-equations constructed from primitives, to formulate the final mathematical expression. We adopt an iterative refinement strategy with a decreasing set of operators. Specifically, we start with a rather rich set of common mathematical operators, and progressively narrow down the operator set according to the obtained results. The final search involves two operators, addition ($+$) and multiplication ($\times$), which align well with related centrality metrics[6,8].

- **Loss.** Since the target $\hat{y}_i = \Theta(\mathbf{X}_i)$ are logits for selecting critical nodes V_c, the absolute values are not as important as the relative order of different nodes. In other words, the mapping from Ω to $\Theta(\mathbf{X})$ is more a binary classification task rather than a regression task. Therefore, we employ the Sigmoid loss function,

$$L_{\mathrm{SR}} = 1 - \tanh(\sigma(\hat{y}_i) \cdot \sigma(\tilde{y}_i)), \tag{34}$$

  where $\hat{y}_i$ is the logit predicted by the DRL model Θ and $\tilde{y}_i$ is the prediction by the mathematical formula from SR.

Evolutionary algorithm. Searching for empirical equations combining primitives ($\Omega = \{\omega_1, \ldots, \omega_P\}$) with mathematical operators ($+$, $\times$) is also an NP-hard problem[45], thus we utilize evolutionary algorithm to efficiently explore the enormous mathematical expression space with the PySR library[46]. Specifically, we keep 60 populations, each of which contains 40 mathematical equations, which evolve towards higher fitting accuracy and lower expression complexity. The evolution contains the following key procedures,

- **Tournament.** It compares a population of mathematical equations, and generally selects the fittest one as the winner of the tournament with large probability. Meanwhile, sub-optimal equations may also be selected through random sampling. Specifically, the probability of selecting the current fittest one $p = 0.9$, and the whole probability sequence over all equations ranked by their fitness is roughly p, $p(1-p)$, $p(1-p)^2$, ... The winners of the tournament are chosen for breeding by the following mutation or crossover procedure.

- **Mutation.** Offspring are generated by mutation which alters the mathematical equations a little bit to search their neighborhood in the mathematical expression space. For instance, we can manipulate one operator or primitive,

$$\text{mutation 1}: \quad \omega_1 \times \omega_2 + \omega_3 \quad \rightarrow \quad \omega_1 \times \omega_2 \times \omega_3 \tag{35}$$
$$\text{mutation 2}: \quad \omega_1 \times \omega_2 \times \omega_3 \quad \rightarrow \quad \omega_1 \times \omega_2 \times \omega_4 \tag{36}$$
$$\text{mutation 3}: \quad \omega_2 \times \omega_3 + \omega_1 \times \omega_4 \quad \rightarrow \quad \omega_1 \times \omega_4 \tag{37}$$

  Other mutations include changing constant terms, generating a replacement equation, and simplifying the equation.

- **Crossover.** Besides unary mutations that occur on one single equation, binary crossovers are performed to mix two winners of the tournament for breeding candidate equations. Specifically, two formulas exchange parts of each other to produce candidate equations, as the following example,

$$\text{crossover}: \quad \omega_1 \times \omega_2 + \omega_3, \quad \omega_2 \times \omega_3 \times \omega_4 \quad \rightarrow \quad \omega_1 \times \omega_2 \times \omega_3 \times \omega_4, \quad \omega_2 + \omega_3 \tag{38}$$

After the evolution consisting of tournament, mutation, and crossover, each formula in the population is simplified and its constant coefficients are optimized. We run 100 iterations of the evolution loop, during the evolution we keep track of a hall of fame H, which records the best fitted equation of different expression complexity so far. The final derived index θ is chosen from H through trading off accuracy and complexity. With the derived θ, we achieve comparable or improved precision in identifying keystone nodes V_c, while ensuring greater interpretability because the mathematical formula is more tangible and human-understandable.

M.6 Refine the discovered equations

A list of candidate equations H is obtained (refer to Supplementary Table 3), which calculates the selection probability of a node based on its corresponding features, allowing us to quantify κ and V_c in a much more explainable manner compared to the original black-box DRL model Θ. The equations generated by SR may appear complex in their raw form, hence we refine these formulas based on dimension and simplify them by removing constant terms, as they do not impact the order of different nodes. Finally, as illustrated in Supplementary Note 4.3 and Supplementary Figure 6c, balancing accuracy and simplicity, we obtain a concise formula denoted as $d \cdot s$, the product of `degree` and `state`, which resides at the core of the black-box RL agent and provides information about the individual contributions of each node to the network's resilience.

M.7 Cellular dynamics

We first investigate the cellular dynamics describing regulatory interactions among genes. The activity of genes are characterized by the following coupled Michaelis-Menten function[2,5,14],

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = -b_i x_i^f + \sum_{j=1}^{N} A_{ij} \frac{x_j^h}{1 + x_j^h}, \tag{39}$$

where self-dynamics $F_i = -b_i x_i^f$ and interactions $G_{ij} = \frac{x_j^h}{1+x_j^h}$ exhibit strong heterogeneity and non-linearity. Specifically, we achieve the dynamical heterogeneity via imposing a power-law to the decay rate b_i,

$$f(b,a) = a \cdot b^{a-1}, \tag{40}$$

where a is a hyper-parameter to control the level of heterogeneity. We set Hill coefficient $h = 2$ for the cooperation level and $f = 1$ for degradation accordingly[2].

To measure network resilience, we first compute the steady states of genes according to equation (39),

$$\mathbf{x}_0 = (x_1, \ldots, x_N)_{t=0} = (10, \ldots, 10) \tag{41}$$

$$\mathbf{x}_T = \mathbf{x}_{t=0} + \int_0^T -\mathbf{b}\mathbf{x}^f + \mathbf{A}\frac{\mathbf{x}^h}{1+\mathbf{x}^h}\mathrm{d}\mathbf{x}, \tag{42}$$

where $\mathbf{b} = (b_1, \ldots, b_N)$ is the heterogeneous dynamic parameters and $T = 400$ to obtain the steady state. Following related literature[2,5], a resilient cellular network indicates positive gene activity while a non-resilient cellular network refers to cell death with no gene activity. Therefore, the resilience function is defined by the following equation,

$$\mathcal{R}(\mathbf{A}, [\mathbf{F}, \mathbf{G}]) = \begin{cases} 1 & \text{if } \langle \mathbf{x}_T \rangle > 0 \\ 0 & \text{if } \langle \mathbf{x}_T \rangle = 0 \end{cases} \tag{43}$$

The resilience function (43) cannot be expressed as explicit mathematical expression, but can only be calculated through integrating the network dynamics as equation (42), which makes the problem (7-11) non-analytic via manual derivations. Please refer to Supplementary Figure 11 for an illustration on system resilience of a cellular network.

M.8 Neuronal dynamics

We then investigate the neuronal dynamics describing the interactions among neurons. The activity of neurons are characterized by the following Wilson–Cowan dynamics[20,21],

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = -b_i x_i + \sum_{j=1}^{N} A_{ij} \frac{1}{1+\mathrm{e}^{\mu - \delta x_j}}, \tag{44}$$

where self-dynamics $F_i = -b_i x_i$ and interactions $G_{ij} = \frac{1}{1+\mathrm{e}^{\mu-\delta x_j}}$ also display strong heterogeneity and non-linearity. The dynamical heterogeneity is achieved similarly by a power-law on b_i as equation (40).

In contrast to cellular dynamics, a neuronal network can feature bifurcation and inactivity under external perturbations, which both indicate that the network loses its resilience[2,5]. Therefore, to comprehensively measure network resilience, we need to compute the steady states involving environmental changes. Specifically, we can compare the steady states obtained from different initial states, where the distinction represents perturbations. Here we include two initial states, low and high, and their corresponding steady states are computed as follows,

$$\mathbf{x}_0^{\mathrm{H}} = (x_1, \ldots, x_N)_{t=0}^{\mathrm{H}} = (10, \ldots, 10), \quad \mathbf{x}_0^{\mathrm{L}} = (x_1, \ldots, x_N)_{t=0}^{\mathrm{L}} = (0, \ldots, 0) \tag{45}$$

$$\mathbf{x}_T^{\mathrm{H}} = \mathbf{x}_0^{\mathrm{H}} + \int_0^T -\mathbf{b}\mathbf{x} + \mathbf{A}\frac{1}{1+\mathrm{e}^{\mu-\delta\mathbf{x}}}\mathrm{d}\mathbf{x}, \quad \mathbf{x}_T^{\mathrm{L}} = \mathbf{x}_0^{\mathrm{L}} + \int_0^T -\mathbf{b}\mathbf{x} + \mathbf{A}\frac{1}{1+\mathrm{e}^{\mu-\delta\mathbf{x}}}\mathrm{d}\mathbf{x}, \tag{46}$$

where $\mathbf{b} = (b_1, \ldots, b_N)$ is the heterogeneous dynamic parameters and $T = 400$ to obtain the steady state. A neuronal network is non-resilient either the two steady states feature bifurcation or the system becomes inactive, otherwise it is resilient under perturbations. Thus the resilience function is defined by the following equation,

$$\mathcal{R}(\mathbf{A}, [\mathbf{F}, \mathbf{G}]) = \begin{cases} 1 & \text{if } \mathbf{x}_T^{\mathrm{H}} = \mathbf{x}_T^{\mathrm{L}} \text{ and } \langle \mathbf{x}_T^{\mathrm{L}} \rangle > 0 \\ 0 & \text{if } \mathbf{x}_T^{\mathrm{H}} \neq \mathbf{x}_T^{\mathrm{L}} \text{ or } \langle \mathbf{x}_T^{\mathrm{L}} \rangle = 0 \end{cases} \tag{47}$$

Similarly, the resilience function (47) is also highly non-analytic, which can only be treated in a computational rather than analytical way. Please refer to Supplementary Figure 12 for an illustration on system resilience of a neuronal network.

M.9 Epidemic dynamics

Finally, we investigate epidemic dynamics, which describe the spread of an infection on a network. We use the classic Susceptible-Infected-Susceptible (SIS) model[22,23], where nodes can be in one of two states: susceptible or infected. The dynamics are characterized by the following function, which describes the probability x_i of a node i being infected:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = -b_i x_i + \sum_{j=1}^{N} A_{ij}(1-x_i)x_j, \tag{48}$$

where the self-dynamics $F_i = -b_i x_i$ represent the recovery process with a recovery rate b_i, and the interactions $G_{ij} = (1-x_i)x_j$ represent the infection process, where a susceptible node $(1-x_i)$ gets infected by its neighbors at an infection rate A_{ij}. We first compute the steady states of nodes according to equation (48),

$$\mathbf{x}_0 = (x_1, \ldots, x_N)_{t=0} = (\varepsilon, \ldots, \varepsilon) \tag{49}$$

$$\mathbf{x}_T = \mathbf{x}_{t=0} + \int_0^T (-\mathbf{b}\mathbf{x} + (1-\mathbf{x}) \circ (\mathbf{A}\mathbf{x}))\mathrm{d}t, \tag{50}$$

where we initialize the system with a small, uniform infection probability ε across all nodes as 0.5 and set T as 400. Following prior study[22], a network is considered non-resilient if the final steady state $\mathbf{x}_T$ is a zero vector, indicating the complete eradication of the infection. Therefore, the resilience function can be calculated as follows,

$$\mathcal{R}(\mathbf{A}, [\mathbf{F}, \mathbf{G}]) = \begin{cases} 1 & \text{if } \langle \mathbf{x}_T \rangle > 0 \\ 0 & \text{if } \langle \mathbf{x}_T \rangle = 0 \end{cases} \tag{51}$$

As with the other dynamical systems, the resilience function (51) is non-analytic and must be evaluated computationally by simulating the system dynamics, making the optimization problem (7-11) intractable via analytical derivations. The system resilience of an epidemic network is similar to the aforementioned cellular network illustrated by Supplementary Figure 11.

M.10 Synthetic network generation

For evaluation, we generate four types of synthetic networks using networkx[47], with number of nodes N ranging from 80 to 200 and an average degree of $\langle d \rangle = 6$. For each type and size, we generate 10 instances with different random seeds. The specific models and parameters are as follows:

- ER and BA networks: Standard implementations where parameters are set to achieve the target average degree.
- Random Partition (RP) networks with communities: Generated with two equal-sized communities ($N/2$). The probability of intra-community edges is set to $2\langle d \rangle / N$, and the probability of inter-community edges is set to $d/5N$ to create a clear community structure.
- Watts-Strogatz small-world networks: Generated with a rewiring probability of 0.4 for each edge.

For calculating the ground-truth values of node removal, we multiply the adjacency matrix $\mathbf{A}$ by a weight factor of $w < 1$ to reduce the number of nodes required for removal to allow for exhaustive search[22] on the weakened network $w \cdot \mathbf{A}$.

M.11 Evaluation protocol for node removal

It is important to clarify the protocol for applying discovered centrality metrics during the node removal process, as the approach differs between our two primary experimental sections due to computational considerations. For the dynamical resilience experiments, we employ an iterative (or adaptive) calculation. After each node is removed, the relevant metrics (e.g., node states s and degrees d) are fully recomputed on the remaining graph. This ensures the attack strategy is dynamic and adapts to the evolving state and structure of the system. For the topological robustness experiments, we use a one-time (or static) calculation. All centrality metrics are computed once on the initial, intact network to generate a fixed ranking of nodes that is used for the entire dismantling sequence. This choice is necessitated by the significantly larger scale of the networks evaluated in this section, for which iterative re-computation would be prohibitively time-consuming.

Regarding feasibility, the computation of $d \cdot s$ is efficient and practical. The degree d is trivially available from the adjacency structure, while s requires only a single simulation round to reach steady state. In many real-world systems, s is directly observable, such as neuron firing rates, gene expression levels, species abundance, or business throughput, eliminating the need

for simulation. In some cases, adaptive computation is also possible by recalculating both quantities after each node removal, though re-evaluating steady state may incur additional cost. For very large-scale systems with incomplete topology data or partial state observations, approximation strategies such as sampling-based degree estimation or surrogate models for state prediction may be necessary.

Regarding metric normalization, we note that the $d \cdot s$ metric is used to rank nodes for keystone node identification, so only the relative ordering among nodes matters rather than absolute values. Normalization does not affect this ordering: if d and s are scaled by positive constants w_1 and w_2, the score $w_1 w_2 \cdot d \cdot s$ preserves the same ranking as $d \cdot s$. Similarly, the network-level indicator $\mathrm{corr}(d,s)$ is invariant to such scaling, since $\mathrm{corr}(w_1 d, w_2 s) = \mathrm{corr}(d,s)$. Therefore, no normalization is imposed in our analysis.

We further clarify that, across the networks and dynamics evaluated in this study, the empirical ranges and variabilities of the structural term d and the dynamical term s are comparable, such that neither term systematically dominates the product $d \cdot s$ and their contributions remain balanced. We therefore do not apply additional normalization in the present experiments. In broader cross-dynamical or cross-network scenarios where d and s may differ substantially in scale, normalization may be needed to maintain a meaningful balance between structural and dynamical contributions. Accordingly, normalization may be assessed when extending the metric beyond the regimes studied here.

Data Availability

The datasets used in the experiments have been made available on GitHub (https://github.com/tsinghua-fib-lab/selinda).

Code Availability

The code used in this research has been made available on GitHub (https://github.com/tsinghua-fib-lab/selinda).

References

1. Buldyrev, S. V., Parshani, R., Paul, G., Stanley, H. E. & Havlin, S. Catastrophic cascade of failures in interdependent networks. *Nature* **464**, 1025–1028 (2010).
2. Gao, J., Barzel, B. & Barabási, A.-L. Universal resilience patterns in complex networks. *Nature* **530**, 307–312 (2016).
3. Albert, R., Jeong, H. & Barabási, A.-L. Error and attack tolerance of complex networks. *Nature* **406**, 378–382 (2000).
4. Fan, C., Zeng, L., Sun, Y. & Liu, Y.-Y. Finding key players in complex networks through deep reinforcement learning. *Nat. machine intelligence* **2**, 317–324 (2020).
5. Sanhedrai, H. *et al.* Reviving a failed network through microscopic interventions. *Nat. Phys.* **18**, 338–349 (2022).
6. Artime, O. *et al.* Robustness and resilience of complex networks. *Nat. Rev. Phys.* 1–18 (2024).
7. Morone, F. & Makse, H. A. Influence maximization in complex networks through optimal percolation. *Nature* **524**, 65–68 (2015).
8. Zhang, Y., Shao, C., He, S. & Gao, J. Resilience centrality in complex networks. *Phys. Rev. E* **101**, 022304 (2020).
9. Grassia, M., De Domenico, M. & Mangioni, G. Machine learning dismantling and early-warning signals of disintegration in complex systems. *Nat. Commun.* **12**, 5190 (2021).
10. Ulgiati, S. & Brown, M. T. Monitoring patterns of sustainability in natural and man-made ecosystems. *Ecol. modelling* **108**, 23–36 (1998).
11. Lalou, M., Tahraoui, M. A. & Kheddouci, H. The critical node detection problem in networks: A survey. *Comput. Sci. Rev.* **28**, 92–117 (2018).
12. Davies, A. *et al.* Advancing mathematics by guiding human intuition with ai. *Nature* **600**, 70–74 (2021).
13. Clusella, P., Grassberger, P., Pérez-Reche, F. J. & Politi, A. Immunization and targeted destruction of networks using explosive percolation. *Phys. review letters* **117**, 208301 (2016).
14. Karlebach, G. & Shamir, R. Modelling and analysis of gene regulatory networks. *Nat. reviews Mol. cell biology* **9**, 770–780 (2008).
15. Barzel, B. & Barabási, A.-L. Universality in network dynamics. *Nat. physics* **9**, 673–681 (2013).

16. Barzel, B., Liu, Y.-Y. & Barabási, A.-L. Constructing minimal models for complex system dynamics. *Nat. communications* **6**, 7186 (2015).

17. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D.-U. Complex networks: Structure and dynamics. *Phys. reports* **424**, 175–308 (2006).

18. Laurence, E., Doyon, N., Dubé, L. J. & Desrosiers, P. Spectral dimension reduction of complex dynamical networks. *Phys. Rev. X* **9**, 011042 (2019).

19. Wu, R.-J., Kong, Y.-X., Di, Z., Bascompte, J. & Shi, G.-Y. Rigorous criteria for the collapse of nonlinear cooperative networks. *Phys. Rev. Lett.* **130**, 097401 (2023).

20. Wilson, H. R. & Cowan, J. D. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. journal* **12**, 1–24 (1972).

21. Wilson, H. R. & Cowan, J. D. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* **13**, 55–80 (1973).

22. Liu, C. *et al.* Deep learning resilience inference for complex networked systems. *Nat. Commun.* **15**, 9203 (2024).

23. Pastor-Satorras, R. & Vespignani, A. Epidemic spreading in scale-free networks. *Phys. review letters* **86**, 3200 (2001).

24. Ren, X.-L., Gleinig, N., Helbing, D. & Antulov-Fantulin, N. Generalized network dismantling. *Proc. national academy sciences* **116**, 6554–6559 (2019).

25. Zdeborová, L., Zhang, P. & Zhou, H.-J. Fast and simple decycling and dismantling of networks. *Sci. reports* **6**, 37954 (2016).

26. Tukamuhabwa, B. R., Stevenson, M., Busby, J. & Zorzini, M. Supply chain resilience: definition, review and theoretical foundations for further study. *Int. journal production research* **53**, 5592–5623 (2015).

27. Schneider, C. M., Moreira, A. A., Andrade Jr, J. S., Havlin, S. & Herrmann, H. J. Mitigation of malicious attacks on networks. *Proc. Natl. Acad. Sci.* **108**, 3838–3841 (2011).

28. Engsig, M., Tejedor, A., Moreno, Y., Foufoula-Georgiou, E. & Kasmi, C. Domirank centrality reveals structural fragility of complex networks via node dominance. *Nat. Commun.* **15**, 56 (2024).

29. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).

30. Page, L., Brin, S., Motwani, R. & Winograd, T. The pagerank citation ranking: Bringing order to the web. Tech. Rep., Stanford infolab (1999).

31. Fawzi, A. *et al.* Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature* **610**, 47–53 (2022).

32. Albert, R. & Barabási, A.-L. Statistical mechanics of complex networks. *Rev. modern physics* **74**, 47 (2002).

33. Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world'networks. *Nature* **393**, 440–442 (1998).

34. Callaway, D. S., Newman, M. E., Strogatz, S. H. & Watts, D. J. Network robustness and fragility: Percolation on random graphs. *Phys. review letters* **85**, 5468 (2000).

35. Romera-Paredes, B. *et al.* Mathematical discoveries from program search with large language models. *Nature* 1–3 (2023).

36. Trinh, T., Wu, Y., Le, Q., He, H. & Luong, T. Solving olympiad geometry without human demonstrations. *Nature* DOI: 10.1038/s41586-023-06747-5 (2024).

37. Senior, A. W. *et al.* Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020).

38. Jumper, J. *et al.* Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).

39. Merchant, A. *et al.* Scaling deep learning for materials discovery. *Nature* 1–6 (2023).

40. Meena, C. *et al.* Emergent stability in complex network dynamics. *Nat. Phys.* 1–10 (2023).

41. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)* (2017).

42. Raffin, A. *et al.* Stable-baselines3: Reliable reinforcement learning implementations. *J. Mach. Learn. Res.* **22**, 1–8 (2021).

43. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

44. Ying, Z., Bourgeois, D., You, J., Zitnik, M. & Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks. *Adv. neural information processing systems* **32** (2019).

45. Virgolin, M. & Pissis, S. P. Symbolic regression is np-hard. *arXiv preprint arXiv:2207.01018* (2022).

46. Cranmer, M. Interpretable machine learning for science with pysr and symbolicregression. jl. *arXiv preprint arXiv:2305.01582* (2023).

47. Hagberg, A., Swart, P. J. & Schult, D. A. Exploring network structure, dynamics, and function using networkx. Tech. Rep., Los Alamos National Laboratory (LANL), Los Alamos, NM (United States) (2008).

Acknowledgements

Funding Statement

Author Contributions

Y.Z., J.D., and Y.L. conceived of the project and designed the research methods. Y.Z. developed the self-inductive symbolized reinforcement learning framework and performed the experiments, with J.D., D.J., J.G., and Y.L. contributing technical advice and ideas. J.D. and Y.L. directed and managed the project. All authors jointly analyzed the results and participated in the writing of the manuscript.

Competing Interests

The authors declare no competing interests.

Tables

Table 1. Evaluation of the discovered formula its corollary formulas of network resilience by our approach compared with existing methods. We report the removal cost on real-world large-scale networks with cellular, neuronal, and epidemic dynamics across different types and sizes. Bold indicates the best overall result; underlining indicates the best result among the baseline methods.

Dynamic	Network Type	N	Analytical Method						ML Method		Topo-Dynamic Method			
			DC[3]	RC[8]	CI[7]	GND[24]	EI[13]	CoreHD[25]	GDM[9]	FINDER[4]	s	PageRank $\cdot s$	RC $\cdot s$	$d \cdot s$
Cellular	Human	355	9	9	14	174	78	37	34	9	**4**	**4**	**4**	**4**
		549	22	23	23	259	171	100	82	22	14	**11**	**11**	**11**
		978	64	64	71	557	470	248	195	67	34	**32**	**32**	**32**
		3125	272	256	293	2026	1114	726	507	263	146	126	127	**125**
	Yeast	491	17	16	18	213	176	101	24	17	**11**	**11**	12	**11**
		701	30	38	32	281	240	141	51	31	9	**8**	**8**	**8**
		931	50	43	61	403	343	209	80	52	19	**16**	**16**	**16**
		1647	121	118	116	808	661	398	193	117	32	**30**	34	**30**
Neuronal	Brain	286	103	105	200	180	258	232	203	205	87	86	86	**84**
		359	158	155	209	236	267	302	251	210	126	126	128	**122**
		676	278	271	476	351	541	570	490	446	225	230	225	**222**
		989	428	430	667	490	889	854	745	543	346	352	347	**335**
Epidemic	Social	69	9	8	11	15	26	24	12	12	**7**	8	**7**	**7**
		133	13	13	40	61	64	51	33	32	11	12	**10**	**10**
		271	34	31	47	110	66	60	51	33	21	18	**17**	**17**
		325	16	15	95	225	21	18	105	82	**4**	**4**	**4**	**4**

Method	Food-Bay-Dry (128)	Food-Bay-Wet (128)	Food-Maayan (183)	Crime (829)	Power-Eris (1176)	HI-II-14 (4165)	Advogato (6539)	DBLP (12591)	Twitter (23370)	Linux (30837)	Facebook (63392)
DC[3]	0.4849	0.4857	0.2438	0.1242	0.1247	0.0691	0.1636	0.1230	0.0091	0.0957	0.3165
CI[7]	0.3868	0.3818	**0.1945**	0.1132	0.0755	0.0561	0.0999	0.0857	0.0092	0.0562	0.2339
GND[24]	0.4502	0.4388	0.3458	0.2570	0.1547	0.3875	0.4404	0.4216	0.3677	0.4491	0.4311
EI[13]	0.4041	0.3984	0.2636	0.3259	0.2614	0.1400	0.2331	0.1759	0.0524	0.2763	0.3632
CoreHD[25]	0.4020	0.3992	0.2641	0.2061	0.1992	0.1155	0.1998	0.1388	0.0152	0.1705	0.3174
PageRank[30]	0.4758	0.4761	0.2277	0.1219	0.0792	0.0652	0.1584	0.1078	0.0090	0.0853	0.2903
DomiRank[28]	0.3907	0.3898	0.2178	0.1166	**0.0509**	0.0591	0.1598	0.1162	0.0090	0.0636	0.3068
GDM[9]	0.4612	0.4543	0.2326	0.1168	0.0748	0.0749	0.1592	0.1088	0.0108	0.0795	0.2994
FINDER[4]	0.4834	0.4846	0.2230	0.1210	0.1029	0.0645	0.1579	0.1106	0.0093	0.0867	0.3095
Ours	**0.3816**	**0.3773**	0.1993	**0.1092**	0.0736	**0.0559**	**0.0987**	**0.0856**	**0.0088**	**0.0499**	**0.2309**

Table 2. Evaluation of the discovered formula of network robustness by our approach compared with existing methods. We report the removal cost on real-world networks (accumulated normalized connectivity (ANC), lower is better). Bold indicates the best result.

Dynamic	Network	N	DC[3]	CI[7]	PageRank[30]	DomiRank[28]	RC[8]: $2\bar{d}+d(d-2\beta)$	Ours: $2\bar{d}+d(\bar{d}-2\beta)$
Cellular	Human	355	9	14	10	9	8	**5**
		978	48	66	48	51	45	**35**
		3125	151	243	180	173	153	**141**
	Yeast	491	17	18	16	17	16	**13**
		931	50	61	54	48	33	**28**
		1647	79	100	101	75	54	**53**
Neuronal	Brain	286	**103**	196	174	196	104	**103**
		676	**267**	465	424	390	269	268

Table 3. Evaluation of the formula discovered by our method in comparison to the state-of-the-art RC formula and baselines on real-world large-scale networks with homogeneous dynamics. Our discovered formula improves over RC by 22.5% and 11.9% in accuracy of keystone node identification for both human and yeast networks driven by cellular dynamics with a maximum improvement of 37.5%, and outperforms RC on two brain networks driven by neuronal dynamics. Bold indicates the best result.

Figure Legends/Captions

Figure 1. The proposed method. a. The proposed self-inductive symbolized reinforcement learning framework. It first dismantles networks with a deep reinforcement learning (DRL) agent searching for keystone nodes, then symbolizes the DRL agent with mathematical formulas unraveling its node selection strategy, leading to the discovery of network resilience metrics. **b. The graph neural network (GNN)-based DRL agent.** It encodes rich features of both network topology and complex dynamics into node representations, based on which it calculates the selection probability of keystone nodes and estimates the network resilience with a policy and value network, respectively. **c. Feature distillation.** It attributes the agent's prediction to individual input features with a differentiable feature selection mask maximizing the mutual information between selected features and model prediction as well as minimizing the number of included features. A sparse set of important features are reserved for further formula discovery. **d. Resilience formula generation.** An evolutionary symbolic regression algorithm is developed to map the distilled important features to keystone nodes identified by the DRL agent with algebraic operators, leading to human-understandable mathematical formulas revealing network resilience.

Figure 2. Node removal performance. a-d. The node removal ratio of different methods (lower means more accurate) for **a.** Erdős–Rényi (ER) networks, **b.** Barabási–Albert (BA) networks, **c.** random partition (RP) networks, and **d.** Watts–Strogatz small-world (SW) networks of growing network sizes, with each size containing 10 synthetic networks driven by gene regulatory dynamics. The compared methods are degree centrality (DC), resilience centrality (RC), Finding key players in complex networks through deep reinforcement learning (FINDER), ranking by node state (State), and our method (Ours). In **a-d.**, boxes extend from the first to the third quartiles, center lines indicate medians, and whiskers extend to the most extreme observations within 1.5 times the interquartile range; outlier points are not shown. Results demonstrate the robustness of our discovered metric, which constantly provides precise identification of keystone nodes, with the number of removed nodes reduced by over 43.4% in comparison to existing approaches. **e-f.** The average error between the node removal ratio and the ground-truth value of different methods for **e.** cellular dynamics and **f.** neuronal dynamics of growing network sizes, with each size containing 10 synthetic ER networks. Results illustrate that existing methods exhibit significant errors as network size increases, with the largest overestimation of keystone nodes by over 7.5%. In contrast, our discovered metric constantly delivers optimal accuracy with near-zero errors to the groundtruth values in removal cost regardless of network size.

Figure 3. Analysis of network-level resilience. a-b. Resilience measured by the number of removed nodes under different correlations between node degree distribution and node state distribution for **a.** neuronal Barabási–Albert (BA) networks and **b.** neuronal Erdős–Rényi (ER) networks. The dynamical parameters are set to different values to achieve different degree-state correlation. Here, d and s denote node degree and node state, respectively, and $\mathrm{corr}(d,s)$ denotes their correlation. The results demonstrate negative relation between degree-state correlation and network resilience, where networks are least resilient under high $\mathrm{corr}(d,s)$ indicating correlated distribution of degree and state, and decreasing $\mathrm{corr}(d,s)$ can lead to resilience improvements of over 183% and 300%. **c-d.** The mass of $d \cdot s$ over different nodes under different degree-state correlation for **c.** neuronal BA networks and **d.** neuronal ER networks. We display the cumulative distribution of $d \cdot s$ in the picture in picture. The mass of $d \cdot s$, which indicates the risk of network collapse, is not concentrated on hub nodes but rather distributed more uniformly across a wide range of nodes under low $\mathrm{corr}(d,s)$, thereby requiring more nodes to be removed to induce the loss of resilience. **e.** $\mathrm{corr}(d,s)$ of real-world ecology networks and supply chain networks. s indicates specie abundance and company throughput for the two categories of networks, respectively. The $\mathrm{corr}(d,s)$ difference between ecology and supply chain networks is over 55.9%, which is consistent with common beliefs that natural systems are more resilience than artificial ones. **f.** $\mathrm{corr}(d,s)$ of real-world supply chain networks of different sizes. Increasing the size of supply chain networks can lead to a 38.8% drop in $\mathrm{corr}(d,s)$, suggesting potential resilience benefits from incorporating diverse additional suppliers.

Figure 4. Resilience prediction across different networks. a-d. Resilience measured by the number of nodes removed required for network collapse and their corresponding degree-state correlation values for the tested four types of **a.** Erdős–Rényi (ER) networks, **b.** Barabási–Albert (BA) networks, **c.** random partition (RP) networks, and **d.** Watts–Strogatz small-world (SW) networks with cellular dynamics. Blue and red denote low and high degree-state correlation, respectively. In **a-d.**, boxes extend from the first to the third quartiles, center lines indicate medians, whiskers extend to the most extreme observations within 1.5 times the interquartile range, and points beyond the whiskers are outliers. **e.** Degree-state correlation of different dynamical parameter setups for three ecological networks. **f.** Resilience measured by the fraction of nodes removed required for network collapse for three ecological networks.