Detecting Vulnerable Nodes in Urban Infrastructure Interdependent Network

Jinzhu Mao
Department of Electronic
Engineering, BNRist,
Tsinghua University,
Beijing, China
maojz22@mails.tsinghua.edu.cn

Liu Cao
Department of Electronic
Engineering, BNRist,
Tsinghua University,
Beijing, China
l-cao20@mails.tsinghua.edu.cn

Chen Gao*
Department of Electronic
Engineering, BNRist,
Tsinghua University,
Beijing, China
chgao96@gmail.com

Huandong Wang
Department of Electronic
Engineering, BNRist,
Tsinghua University,
Beijing, China
wanghuandong@tsinghua.edu.cn

Hangyu Fan
Tsingroc Inc.,
Beijing, China
fanhangyu@tsingroc.com

Depeng Jin
Department of Electronic
Engineering, BNRist,
Tsinghua University,
Beijing, China
jindp@tsinghua.edu.cn

Yong Li
Department of Electronic
Engineering, BNRist,
Tsinghua University,
Beijing, China
liyong07@tsinghua.edu.cn

ABSTRACT

Understanding and characterizing the vulnerability of urban infrastructures, which refers to the engineering facilities essential for the regular running of cities and that exist naturally in the form of networks, is of great value to us. Potential applications include protecting fragile facilities and designing robust topologies, etc. Due to the strong correlation between different topological characteristics and infrastructure vulnerability and their complicated evolution mechanisms, some heuristic and machine-assisted analysis fall short in addressing such a scenario. In this paper, we model the interdependent network as a heterogeneous graph and propose a system based on graph neural network with reinforcement learning, which can be trained on real-world data, to characterize the vulnerability of the city system accurately. The presented system leverages deep learning techniques to understand and analyze the heterogeneous graph, which enables us to capture the risk of cascade failure and discover vulnerable infrastructures of cities. Extensive experiments with various requests demonstrate not only the expressive power of our system but also transferring ability and necessity of the specific components. All source codes and models including those that can reproduce all figures analyzed in this work are publicly available at this link: https://github.com/tsinghua-fib-lab/KDD2023-ID546-UrbanInfra.

CCS CONCEPTS

• **Information systems** → **Data mining**;

KEYWORDS

Urban Infrastructure Network; Interdependent Network; Graph Neural Networks; Reinforcement Learning

1 INTRODUCTION

Urban infrastructure [6] refers to the engineering facilities essential for the regular running of cities, including electricity supply, transportation, and communication, etc. These urban infrastructures exist naturally in the form of networks, including electricity networks, road networks, communication networks, etc. Further, various infrastructure networks are coupled and interdependent, forming a typical system of interdependent networks [3, 4]. Moreover, cities confront the danger of natural catastrophes such as earthquakes, typhoons, and rainstorms, which often destroy functional units of different urban infrastructures. Due to the interplay between different infrastructures and their functional reliance, destruction often extends beyond the affected area, leading to widespread paralysis
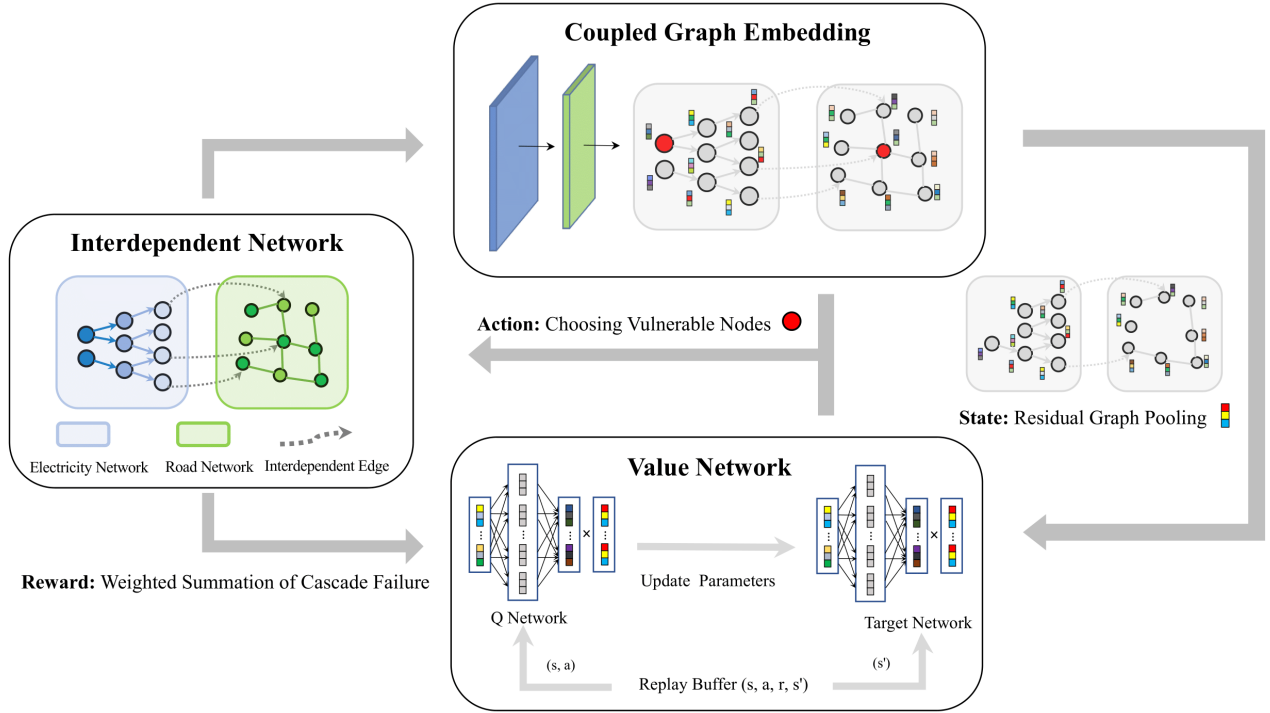
Figure 1: Illustration of our proposed system.

of urban infrastructure. This exacerbates the vulnerability of urban infrastructures [3, 4]. Urban infrastructures are essential for the proper functioning of various services in a city, including housing and production, and are closely tied to the well-being of its citizens. Therefore, gaining a profound understanding of the vulnerability of urban infrastructure holds significant value.

However, understanding and characterizing the vulnerability of urban infrastructure is a difficult task. First, networks of different urban infrastructures exhibit diverse topological structures [9, 10, 28], which differ in terms of cluster structure, cycle structure, degree distribution, centrality, etc. The high-dimensional topological structure and the diverse topological features make it challenging for knowledge-based expert systems [18] or feature engineering methods [1] to thoroughly characterize it, which is the first challenge. Second, there is a strong correlation between these topological characteristics and infrastructure vulnerability [5, 14]. The states of different infrastructure networks have complicated evolution mechanisms. For example, the evolution of vehicle distribution on road networks is driven by the interaction between traffic control signals and driver actions, while the evolution of the distribution of power flow in electricity networks is driven by Ohm's law. These evolution mechanisms compound the functional dependencies of the infrastructure network to exacerbate the higher-order propagation effects of damage. How to model the complicated effect of evolution mechanisms of urban infrastructures is the second challenge.

In this paper, we focus on modeling the vulnerability of interdependent networks composed of diverse urban infrastructures using a data-driven approach. We develop a graph convolutional

neural network, which can efficiently and exhaustively extract the topological features of each infrastructure network as well as the topological features of the interdependency between them, via optimizing the loss function of graph reconstruction. Further, we construct a simulator of the interdependent networks composed of urban infrastructures and develop a reinforcement learning (RL) module by interacting with it. This module deeply models the network evolution mechanism by using the influence of damaged functional units as a reward, which can accurately characterize the vulnerability of the system.

The contributions of this work can be summarized as follows:

- We propose a graph convolutional neural network, which performs a coupled graph construction through semi-supervised embedding learning and can efficiently and exhaustively extract the topological features of each infrastructure network as well as the interdependency between them.
- We construct a reinforcement learning (RL) module, which makes full use of diverse graph embeddings. It can discover a set of vulnerable functional units of infrastructures in the specified reward setting, thus accurately characterizing the vulnerability of the system.
- Extensive experimental results show that our proposed model can effectively characterize the vulnerability of the system while interacting with an urban simulator. Specifically, the influence of the damaged functional units selected by our proposed algorithm significantly beat those of state-of-the-art algorithms.

Table 1: Notations.

Symbols	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Coupled graph $\mathcal{G}$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges
$\mathcal{G}^e = (\mathcal{V}^e, \mathcal{E}^e)$	Electric network $\mathcal{G}^e$, where $\mathcal{V}^e$ is the set of power stations and $\mathcal{E}^e$ is the set of wires
$\mathcal{E}'$	Set of interdependent directed edges
$\mathbf{f}_v, \mathbf{h}_v^l, \mathbf{f}_v$	Initial, process and final embedding of node v
$\mathcal{N}(\cdot)$	Set of neighbor nodes of given one
L	Iteration number or depth of GNN
$\mathbf{W}^i$	Weight matrix of graph neural network at depth i
$\mathcal{G}^n, \mathcal{G}^p$	Negative graph and positive graph
$\mathcal{S}$	Set of states
$\mathcal{A}$	Set of actions
$\mathcal{P}(s' \mid s, a)$	Transition function, where s' is the state at next step
$\mathcal{R}(s, a)$	Reward function
γ	Discount factor
v_k, s_k, r_k	Selected node, state and reward at step k
$Q(s, v)$	Action-value function
$P(v^e, \mathcal{G}^e)$	Function that can calculate the decreased power of $\mathcal{G}^e$ after v^e is damaged
$E(v^e, \mathcal{G})$	Function that returns a set of invalid traffic lights affected by damaged v^e
$A(v^r, \mathcal{G}^r)$	Function that can calculate the decreased value of connectivity of $\mathcal{G}^r$ after v^r is damaged
a^e, a^r	Weight coefficients of $\mathcal{G}^e$ and $\mathcal{G}^r$

2 PROBLEM STATEMENT

Starting from the basic interdependent network, we select two closely connected infrastructure networks – road network [7] and electricity network [29] from the city. The road network consists of a multitude of roads and intersections, representing edges and nodes, respectively. and the electricity network is a tree-like structure digraph composed of a large number of power stations at different levels, such as 220 kV, 110 kV, and 10 kV power stations. These two networks are connected by traffic lights located at intersections and low-level (10kV) power stations. In the road network, the state of traffic lights directly affects the connectivity of the road network. Specifically, if a traffic light stops working due to damage or power failure, roads it connects will become chaotic and even crowded, and more roads could be affected over time. As for the electricity network, it transmits electricity level by level to infrastructures of other networks. For instance, the failure of a 220 kV power station will cascade to 110 kV power stations, resulting in the shutdown of certain 10 kV power stations, ultimately impacting the functioning of traffic lights linked to those 10 kV power stations. Therefore, there are three states for each infrastructure: normal, damaged, and invalid, and "invalid" means that a node stops working because the nodes it connected are damaged or invalid.

To represent the interdependent network, we consider a coupled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that consists two different structural graphs: the road network $\mathcal{G}^r = (\mathcal{V}^r, \mathcal{E}^r)$ and the electricity network $\mathcal{G}^e = (\mathcal{V}^e, \mathcal{E}^e)$. $\mathcal{V} = \mathcal{V}^r \cup \mathcal{V}^e$ is the set of nodes, $\mathcal{E} = \mathcal{E}^r \cup \mathcal{E}^e \cup \mathcal{E}'$ is the set of edges, where $\mathcal{E}' = \{e_{v^e v^r}, \forall v^e \in \mathcal{V}^e, \forall v^r \in \mathcal{V}^r\}$ is the set of interdependent directed edges between power stations and traffic lights. A notation table is organized to make this paper easier to understand shown in Table 1. We expect to discover a set of nodes from the interdependent network as the vulnerable nodes, that once they are damaged, the coupled graph will be greatly affected, where "affected" depends on scene and purpose when making decisions. For instance, if we pay more attention to the condition of road network, the number of crowded roads can serve as the primary indicator of "affected"; if we care more about the state of electricity network, the power decrease in the grid could be the main indicator of "affected".

3 THE PROPOSED SYSTEM

In the course of detecting the vulnerable nodes in the coupled graph, based on the strong representation ability of graph neural network (GNN) for graph-structured data and the ability of deep reinforcement learning (DRL) to solve sequential decision-making problem, we devise a new system to deal with this task. Figure 1 illustrates the architecture of our proposed system.

3.1 Coupled Graph Representation

We expect to take into account the direct and indirect information of the graph while choosing nodes, but the state of the graph can be very complex. Traditional methods[23, 37] usually use fixed features to represent graphs, such as degree distributions, clustering coefficients, and so on. Due to the complex evolutionary mechanism of the coupled graph, these methods are not applicable to most coupled graphs and perform poorly when they characterize the graph's structural information. Based on graph neural networks, we leverage semi-supervised embedding learning [40] to obtain the coupled graph embedding that could extract the node features and graph topology in a low-dimensional embedding space.

3.1.1 Graph Construction. Given an urban interdependent network consisting of an electricity network and a road network, we represent it as a coupled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ according to the previous setting. We map each node $v \in \mathcal{V}$ to a initial d-dimensional vectors $\mathbf{f}_v$ and define a node embedding matrix $\mathbf{F}_v = \{\mathbf{f}_v, \forall v \in \mathcal{V}\} \in \mathbb{R}^{d \times |\mathcal{V}|}$ as,

$$\mathbf{f}_v = \begin{cases} \mathbf{f}_v^e, & \text{if node } v \in \mathcal{V}^e; \\ \mathbf{f}_v^r, & \text{if node } v \in \mathcal{V}^r. \end{cases} \quad (1)$$

Note that the initial embeddings $\mathbf{f}_v^e$ and $\mathbf{f}_v^r$ are pre-trained from the respective graphs by the method described below, while the initial embeddings of them are randomly set. To improve its characterization ability, we use graph neural networks[13] to incorporate node attributes and capture the coupled graph structural information and other unseen data. On account of initial embeddings that have already contained the information of subgraphs, we are able to explore more information about the interdependent relationship in the coupled graph.

3.1.2 Graph Neural Network. Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and initial embeddings of all nodes $\{\mathbf{f}_v, \forall v \in \mathcal{V}\}$ are provided as input. Let l denote the depth and $\mathbf{h}_v^l \in \mathbb{R}^{d \times 1}$ denotes the embedding vector for node v at step l, each node $v \in \mathcal{V}$ aggregates the embeddings of its immediate neighbor nodes, $\left\{\mathbf{h}_u^l, \forall u \in \mathcal{N}(v)\right\}$, into a single vector $\mathbf{h}_{\mathcal{N}(v)}^l \in \mathbb{R}^{d \times 1}$, where $\mathcal{N}(\cdot)$ stands for the set of neighboring nodes. Note that $\mathbf{h}_v^0$ is defined as the input features $\mathbf{f}_v$. The aggregation process is shown as follows,

$$\mathbf{h}_{\mathcal{N}(v)}^l = \text{AGG}\left(\left\{\mathbf{h}_u^l, \forall u \in \mathcal{N}(v)\right\}\right), \quad (2)$$

$$\mathbf{h}_v^{l+1} = \sigma\left(\mathbf{W}^l \cdot \text{MEAN}\left(\left\{\mathbf{h}_v^l, \mathbf{h}_{\mathcal{N}(v)}^l\right\}\right)\right), \quad (3)$$

where AGG is a summation function that aggregates information from local neighbors [22], σ stands for the rectifier function ReLU, MEAN is the mean function, and $\mathbf{W}_1^l \in \mathbb{R}^{d \times d}$ is a weight matrix to be learned at step $l + 1$. By defining the depth or iterations
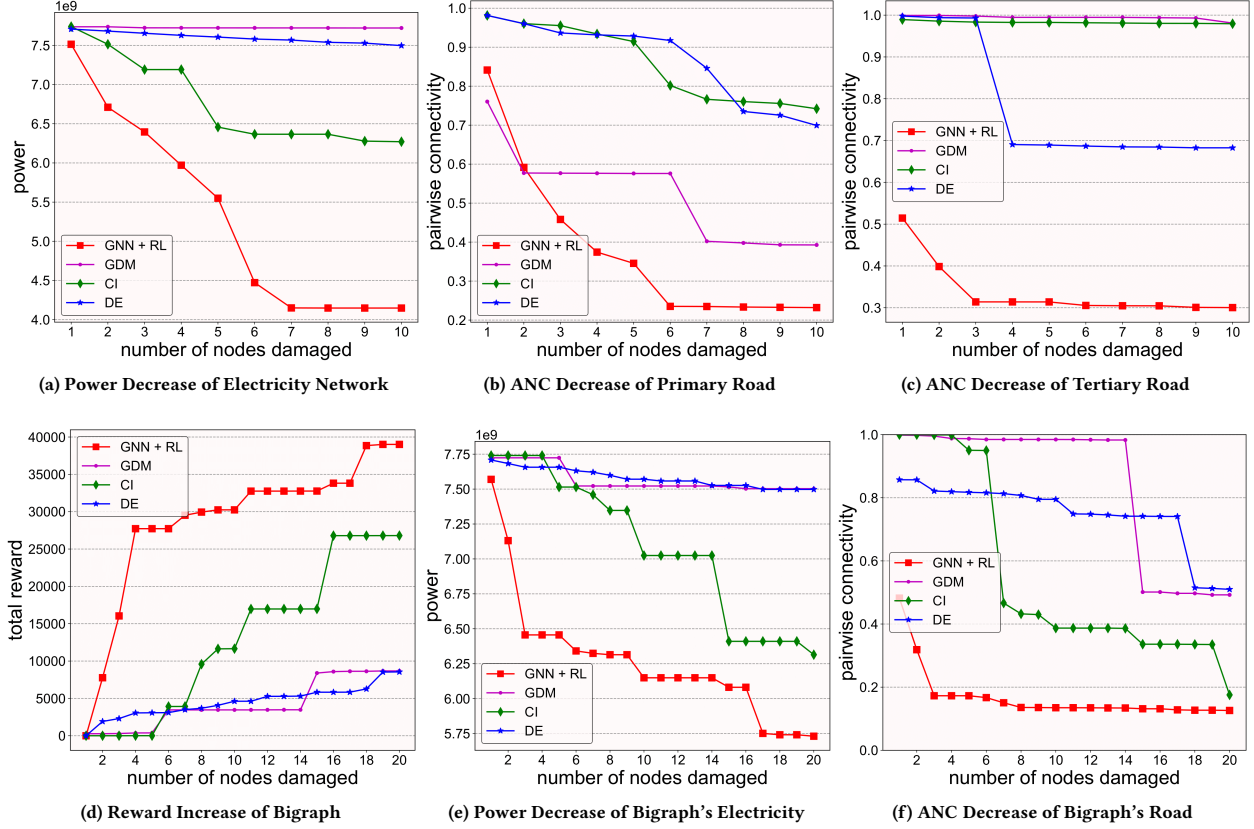
(a) Power Decrease of Electricity Network (b) ANC Decrease of Primary Road (c) ANC Decrease of Tertiary Road

(d) Reward Increase of Bigraph (e) Power Decrease of Bigraph's Electricity (f) ANC Decrease of Bigraph's Road

Figure 2: Performance of different methods on different networks.

L, we can get the final representation for all nodes at depth L, $\left\{ \mathbf{z}_v \equiv \mathbf{h}_v^L, \forall v \in \mathcal{V} \right\}$, which contains L-order neighbor nodes' information.

3.1.3 Semi-supervised Embedding Learning. There are cascade relationships between infrastructures that are not connected directly in the urban network, and the topology of a network implicitly reveal the signal of similarity and correlation among nodes, which can be used as supervised signals to obtain the representation vectors. Therefore, the task can be treated as a link prediction [16] problem, which aims to estimate the likelihood of the edges between nodes, based on observed edges and attributes of nodes. Then we take all of the edges in the graph as a positive set and sample several edges that do not exist as a negative set, which forms a positive graph $\mathcal{G}^p$ and a negative graph $\mathcal{G}^n$, respectively. To learn parameters of the aggregator function and weight matrices $\left\{ \mathbf{W}^l, \forall l \in \{1, \ldots, L\} \right\}$ via stochastic gradient descent, we firstly define a link prediction function as follows,

$$S(\mathbf{Z}_v, \mathcal{G}) = \left\{ \mathbf{z}_v^T \cdot \mathbf{z}_u, \forall e_{vu} \in \mathcal{E} \right\}, \quad (4)$$

which calculates the inner product of embedding vectors for two nodes on each existing edge as predicted weight, on given graph $\mathcal{G}$. Then we apply a margin loss function [30] to the node embeddings

$\mathbf{Z}_v$ as follows,

$$J_{\mathcal{G}} (\mathbf{Z}_v) = \text{MEAN}(\max(0, M - S(\mathbf{Z}_v, \mathcal{G}^p) + S(\mathbf{Z}_v, \mathcal{G}^n)) + \lambda \|\Theta\|^2, \quad (5)$$

where M is a constant parameter, the second term of the loss function performs $L2$ regularization where Θ stands for model parameters and λ controls the penalty strength. For $\mathcal{G}^e$ and $\mathcal{G}^p$, the weights simply represent the presence or absence of edges, and for coupled graph, $\mathcal{G}$, weights are set to even indicate the type of the edges. While unknown edges are predicted by the known edges, we can obtain more superior embeddings for the interdependent network with cascading relationships, through semi-supervised learning [35].

3.2 Vulnerable Nodes Detecting

We obtain a coupled graph embedding matrix through a graph convolutional neural network, which can efficiently extract the topology features of the interdependent network. To make full use of this information, and find a set of vulnerable nodes in the network, we construct a reinforcement learning module. Reinforcement learning [31] is a sequential decision process where the agent is trained to take optimal actions for different scenarios of an environment. We formulate the RL problem as a Markov decision process (MDP), which is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ with a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a reward function $\mathcal{R}(s, a)$, the transition function
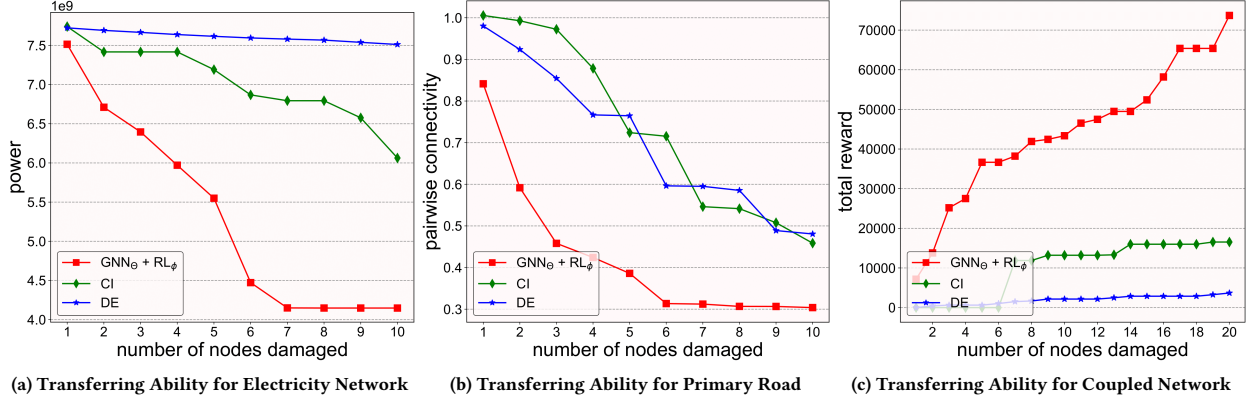
(a) Transferring Ability for Electricity Network   (b) Transferring Ability for Primary Road   (c) Transferring Ability for Coupled Network

Figure 3: Transferring ability for different Networks.

$\mathcal{P}(s' \mid s, a)$ and a discount factor γ, $s, s' \in \mathcal{S}, a \in \mathcal{A}$. $\mathcal{P}(s' \mid s, a)$ governs the joint probability distribution $p(s' \mid s, a)$ of transitioning to new state s' after taking action a in state s. Given a policy π, which fully determines the agent' behavior, the action-value function $Q_\pi(s, a)$ is defined as the expected reward when starting from s, taking action a, and following the policy π. In our proposed system, the action is to choose vulnerable nodes, the state is the residual coupled graph being analyzed after chosen nodes are damaged and the reward is the resulting weighted summation of cascade failure from the environment which represents the given interdependent network.

3.2.1 Framework. Given the couple graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the corresponding pre-trained node embeddings $\mathbf{Z}_v \in \mathbb{R}^{d \times |\mathcal{V}|}$ as the input of our model, we expect to sequentially select K vulnerable nodes from the graph as output for a complete epoch. We use the node embedding matrix $\mathbf{Z}_v$ to represent the universal state s of the current network (environment), and the initial state can be defined as the graph pooling, $\mathbf{s}_1 = \text{MEAN}(\mathbf{Z}_v) \in \mathbb{R}^{d \times 1}$. Then at each step k, the agent selects a node $v_k \sim \pi(\cdot \mid \mathbf{s}_k)$ as the action, where π is the policy. At the same time, next state $\mathbf{s}_{k+1}$ is calculated by removing node v_k as follows,

$$\mathbf{s}_{k+1} = \text{MEAN}(\mathbf{Z}_v \backslash \{v_1, \dots, v_k\}), \quad (6)$$

where $\mathbf{Z}_v \backslash \{v_1, \dots, v_k,\}$ is the residual node embeddings after removing embeddings of nodes in the set $\{v_1, \dots, v_k\} \subseteq \mathcal{V}$. Then the environment judges whether selected node v_k is a power station or a junction, and set it damaged in the urban network to obtain the corresponding reward r_k. The agent update its policy $\pi(v_k \mid \mathbf{s}_k)$ with reward r_k, and choose the next node v_{k+1}. Continue this process until epoch ends, and a trajectory $\tau = (\mathbf{s}_1, v_1, r_1, \dots, \mathbf{s}_K, v_K, r_K)$ is determined by the agent under the policy π. The learning objective is to discover an optimal policy π^* that can maximize the total reward for this whole trajectory: $\sum_{k \geq 1} \gamma^k r_k$. Based on *Bellman equation* [31], the optimal action-value function $Q_{\pi^*}(s, v)$ can be derived as:

$$Q_{\pi^*}(\mathbf{s}, v) = \sum_{s' \in \mathcal{S}} p(s' \mid \mathbf{s}, v) \left[\mathcal{R}(v) + \gamma \max_{v'} Q^*(\mathbf{s}', v') \right], \quad (7)$$

A common method of obtaining a new policy for an action-value function is to adopt ϵ-greedy policy [17], which means that the agent either chooses the node with the highest value with probability $1 - \epsilon$ or chooses a node randomly with probability ϵ. It has been proved that the agent following ϵ-greedy policy could obtain the maximum total rewards when $Q_\pi(s, v)$ is optimal [32].

3.2.2 Value Network and Optimization. Owing to the agent's action that choosing a node at each step is discrete, and the action space is extremely large we design our vulnerable nodes detecting model inspired by Deep Q-Learning Network (DQN) algorithm [26], which takes advantage of a convolution neural network to approximate the $Q_\pi(\mathbf{s}, v)$. To make full use of the convolution neural network and structural information learned by the node embeddings $\mathbf{Z}_v$ when choosing nodes, the action-value function $Q_\pi(\mathbf{s}, v)$ are defined as follows,

$$Q_\pi(\mathbf{s}_k, v_k; \Theta) = \text{MEAN}((\theta_2 \cdot (\sigma(\theta_1 \cdot \mathbf{Z}_v)))^T \cdot [\mathbf{s}_k, \dots, \mathbf{s}_k]_{|\mathcal{V}|}), \quad (8)$$

where $\theta_1 \in \mathbb{R}^{2d \times d}$ and $\theta_2 \in \mathbb{R}^{d \times 2d}$ are the optimized parameters, σ stands for a rectifier function and $[\mathbf{s}_k, \dots, \mathbf{s}_k]_{|\mathcal{V}|}$ is a function that concatenating $\mathbf{s}_k$ for $|\mathcal{V}|$ times. At each step, $Q_\pi(\mathbf{s}_k, v_{k-1})$ is used for the ϵ-greedy selection of the next action v_km, and the agent stores the transition tuple $(\mathbf{s}_k, v_k, r_k, \mathbf{s}_{k+1})$ into an experience replay buffer, the size of which is E. The parameters Θ are trained over tuples sampled uniformly from the reply buffer via stochastic gradient to minimize the mean-squared loss descent :

$$J_Q(\Theta) = \text{MEAN}(\| (r_k + \gamma \max_v \hat{Q}(\mathbf{s}_{k+1}, v; \hat{\Theta})) - Q(\mathbf{s}_k, v_k; \Theta) \|^2), \quad (9)$$

where $\hat{\Theta}$ represent the parameters of the target net $\hat{Q}$, which only updates after several iterations by copying Θ of evaluation net Q. This approach ensures a more stable and efficient training process. Since the graph is extracted from real data, we obtain the reward r by simulating the network changes after chosen nodes are damaged. To detect the vulnerable infrastructures in the road network, given the chosen node v_k, reward r is the weighted summation of the decreased power of the electric network and decreased value of connectivity of the road network. At step k, the reward function

$\mathcal{R}(v_k)$ are defined as follows,

$$\mathcal{R}(v_k) = \begin{cases} a^e P(v_k, \mathcal{G}^e) + a^r A(E(v_k, \mathcal{G}), \mathcal{G}^e), & \text{if node } v_k \in \mathcal{V}^e; \\ a^r A(v_k, \mathcal{G}^e), & \text{if node } v_k \in \mathcal{V}^r; \end{cases}$$ (10)

where a^e and a^r are weight coefficients, $P(v_k, \mathcal{G}^e)$ is a function that can obtain the decreased power of given electricity network $\mathcal{G}^e$ after power station v_k^e is damaged by Power Flow Calculation, $E(v_k, \mathcal{G})$ is a function that returns a set of invalid traffic lights affected by damaged power station v_k^e and $A(v_k, \mathcal{G}^e)$ is a function that can calculate the decreased value of connectivity of the road network. We can set a reasonable reward by adjusting the weight coefficients, to detect the ideal vulnerable nodes on the coupled graph.

4 EVALUATION

In this section, we aim to evaluate the proposed model with extensive answers to the following research questions (RQs).

- **RQ1:** Could our proposed method achieve the best performance compared with existing methods?
- **RQ2:** What about the transferability of our model? Specifically, how about the model's performance when transferred into a new interdependent network?
- **RQ3:** What about the effectiveness of each component in the proposed method?
- **RQ4:** Why can our method find the vulnerable nodes in the graph? What about the failure process in the interdependent network?

4.1 Experimental Settings

4.1.1 Experimental environment. For the single network, we leverage the electricity network $\mathcal{G}^e$ with 10,887 nodes and 11,438 edges, including different levels of power stations such as 550kV and 220kV, and infrastructure such as traffic lights. In addition, two different road networks—primary road $\mathcal{G}^{rp}$ with 1,035 nodes and 1,161 edges, and tertiary road $\mathcal{G}^{rt}$ with 4,825 nodes and 5,025 edges included, are also applied the model introduced before to choose vulnerable nodes, using metrics followed in Section 4.1.2. For the interdependent network with 15,712 nodes and 21,191 edges, representing the coupled electricity network and the tertiary road we attack nodes consisting of two graphs, importing the basic rules that once a power station fails, the corresponding traffic lights stop working.

4.1.2 Metrics. We use the representative metrics to evaluate the selection performance of the proposed model and the existing method as follows.

- **Electricity Network:** To assess the impact on the electricity network, we calculate the current power after each node is damaged or disabled using flow calculation within the electricity network.
- **Road Network:** For the road network, we use ratio of connectivity [11], which is defined as follows,

$$\sigma(\mathcal{G}) = \sum_{C_i \in \mathcal{G}} \frac{\delta_i(\delta_i - 1)}{2},$$ (11)

where C_i is the ith connected component in the current graph $\mathcal{G}$, and δ_i is the size of C_i. Another metric for the road network is

the size of GCC, which refers to the giant connected component, calculated by:

$$\sigma_{gcc}(\mathcal{G}) = \max\{\delta_i; C_i \in \mathcal{G}\}.$$ (12)

- **Coupled Network** For the coupled network, we use the reward defined as the weighted summation of power or connectivity decrease after each node is damaged in the electricity network or road topology correspondingly, to jointly evaluate the performance on two networks.

4.1.3 Baselines. We compare our proposed method with the following competitive baselines.

- **DE [19, 37]:** Here we represent the degree of node v_i as d_i. This method sorts the degree d_i of each node v_i in the graph and chooses the greatest ten nodes to be removed sequentially.
- **CI [27]:** The Collective Influence (CI) measure is defined as the product of the node's reduced degree and the sum of the reduced degrees of the nodes that are within a constant hop away from it. This method is calculated by $CI(v) = (d_v - 1) \sum_{u \in \mathcal{N}(v)} (d_u - 1)$, for each node, which describes the proportion of other nodes that can be reached from a given node, assuming the nodes with higher CI values play more crucial roles in networks. The CI method sequentially removes the node with the highest CI value and recalculates the CI values of the remaining nodes for the rest following operations.
- **GDM [15]:** Graph Dismantling with Machine Learning (GDM), utilizes a supervised learning approach to dismantle the entire network. To generate the training data and labels, we randomly sample nodes from the complete set and classify them into positive and negative sets. We then calculate the ratio of nodes belonging to the positive set for those that exist in both sets. The pre-trained node embeddings obtained from a graph neural network are used as inputs to a neural network, such as a Multilayer Perceptron (MLP), for the supervised learning process. The neural network computes outputs for all nodes, which are subsequently sorted in descending order to identify nodes for removal as a comparison to our method.

4.2 Overall Performance Comparison (RQ1)

We first present the overall performance comparison in Figure 2. In figure **(a)-(c)** we demonstrate the testing results on single networks including one electricity network and two road networks. In figure **(d)-(e)** we demonstrate the testing results on coupled network consisting of electricity network and tertiary road network. Based on the results, we have the following observations.

- **Performance varies by different baselines.** Based on observations, we have noticed distinct performance differences among various baselines when applied to single and coupled networks. In the electricity network, CI identifies nodes as more vulnerable compared to DE due to the clustering distribution of power stations. Specifically, a 110kV station typically has only one 220kV upstream connection but multiple 10kV downstream connections. In contrast, DE performs better than CI in road networks, specifically primary and tertiary roads, due to the degree homogeneity, where road nodes typically have one incoming and one outgoing connection. Comparing the heuristic methods in the coupled network, CI still outperforms DE. We have established a
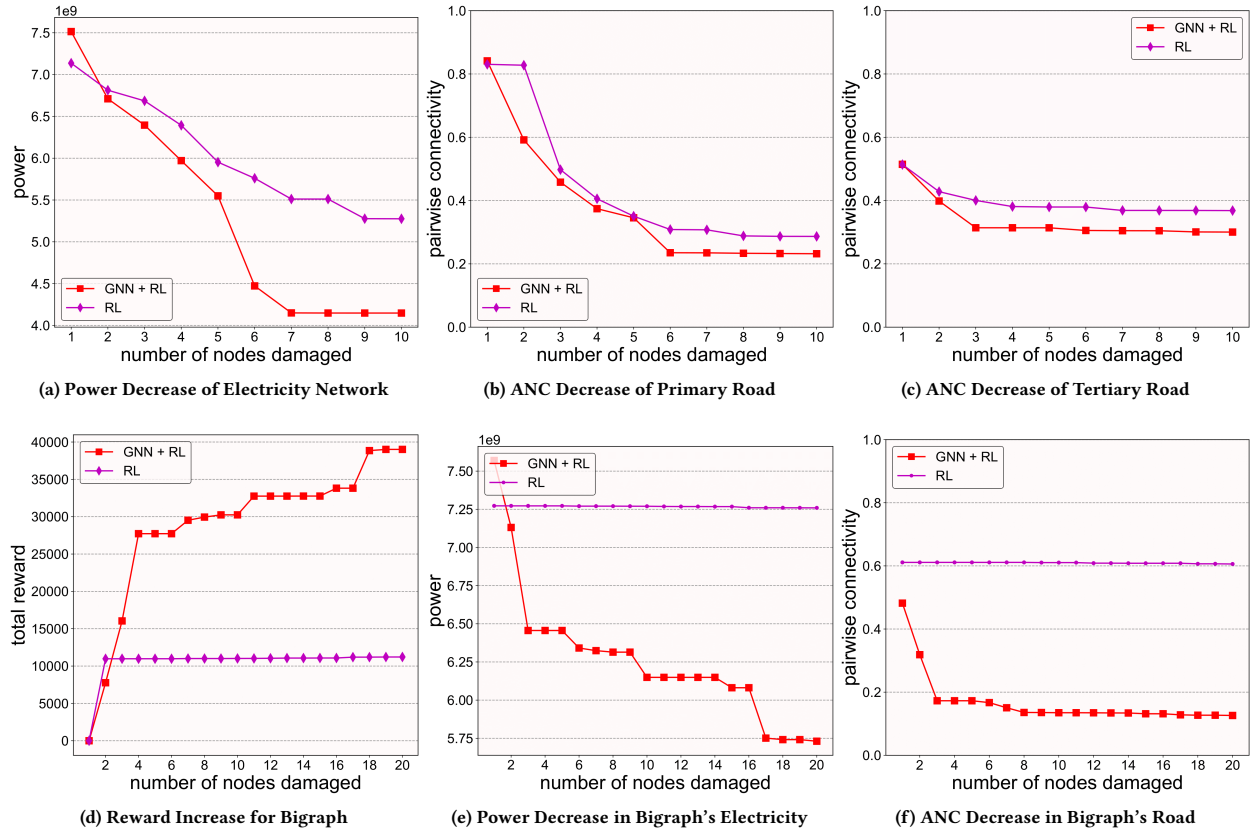
Figure 4: Ablation experiments for different networks.

cascade rule where a power station failure leads to the shutdown of traffic lights supported by that station, resulting in congestion and chaos. This demonstrates that power stations have a greater influence in this coupled network. Regarding machine-assisted analysis, the method works better than heuristic approaches in road networks because it requires fewer enabled nodes to cover a majority of nodes using a set of samples. However, in the electricity network and coupled network, sampling from such a large number of nodes struggles to cover even half of the total nodes. This poses a challenge due to the significant disparity between the volume of testing data and the available training data, resulting in unsatisfactory performance.

- **Our method achieves the best performance.** In comparison to heuristic methods and other learned machine approaches, our method consistently identifies critical nodes to minimize the current state and maximize the defined reward. By damaging fewer than ten nodes out of a total of 10,887 in the entire graph, our method reduces the electricity power to approximately 50% and even lower in the case of road networks. The power decrease achieved by our method is 50% lower than that of CI, the connectivity obtained is approximately 75% lower in primary roads and about 43% lower in tertiary roads compared to the best-performing alternative methods. Additionally, the accumulated

reward in our process is approximately 37% higher than that of the CI method.

- **Our method gets the result fast.** Our method efficiently simulates the entire environment and rapidly identifies vulnerable nodes, yielding a stable and effective solution within approximately 500 epochs.
- **Cascade failure exists.** Cascade failure does exist between the topology of interconnectedness between infrastructures. Without prior knowledge, our method could understand and characterize such a cascading rule driven by the data itself and model the complicated effect of evolution mechanisms of urban infrastructures.

4.3 Study on Transferring Ability (RQ2)

As mentioned before, one of the primary purposes is to characterize the vulnerability of the urban system to descend the influence of damaged functional units in cities. However, inaccuracy is always a property of real-world data and the infrastructures will also be continuously built and renovated. Due to the time and computation costs to re-train a new agent finding vulnerable nodes, we try to get the node embedding that responds to the real-world fast and accurately, and leverage the original parameters of value networks to find nodes and protect the current system, to study the transferring ability of our framework [33]. The detailed setup of the transferability experiments is provided in Appendix. The results

have been shown in Figure 3, figure **(a)-(c)** show the experimental results on the electricity network, primary road, and coupled network, respectively. CI can still find fragile nodes in the electricity and road network. Compared with baselines methods on the new-generated topologies, our transferring system still works better than heuristic methods including DE and CI, obtaining about 46% lower power decrease and 75% lower connectivity than the best baseline correspondingly and reaching a plateau after a small set of nodes damaging. In the coupled network, the total reward obtained by our method is about 20 times the reward of DE and 3.4 times the reward of CI. In conclusion, our method has very strong transferring ability and robustness enabling us to avoid huge costs of time and computation. In addition, the improvement is meaningful and evident as urban infrastructures change rapidly in the real world nowadays.

4.4 Ablation Study (RQ3)

In reinforcement learning, the agent's objective is to maximize the expectation of the total reward by exploring the environment and taking appropriate actions. As reinforcement learning algorithms become more powerful, agents can handle increasingly complex environments. However, when dealing with heterogeneous graphs consisting of different types of nodes and edges, the exploration-exploitation process for the agent can become challenging. Our hypothesis suggests that providing the agent with a confusing or randomly generated environment, specifically random-generated node embeddings, may result in a longer training process to find a stable solution, and the obtained results may be unsatisfactory.

Using the same datasets and metrics as described in Section 4.2, we compare the training duration and final results between our RL agent trained with pre-trained node embeddings generated by GNN and those generated randomly. The comparison is presented in Figure 4, following the same layout as Figure 2. As indicated in the dataset introduction, the performance of the GNN + RL approach outperforms the learning process in a random environment. Notably, the performance gap between the two approaches is more pronounced in the case of the coupled network compared to individual networks. Additionally, the ablation study on the two road networks shows a subtle difference since road structures are simpler and more limited, allowing for similar performance when the training process is extensive. However, in the interdependent network with multiple types of nodes and edges, learning valuable information from random representations becomes challenging. This results in poor performance during testing, where the total reward plateaus and the metrics of the two interacting single networks show a downward trend.

In conclusion, we have confirmed the validity of our assumption. The inclusion of the GNN component in our framework is essential and significantly enhances the performance of the agent when dealing with graph environments, particularly heterogeneous graphs.

4.5 Case Study (RQ4)

Figure 5 visualizes the effects of a series of node damages, displaying three sets of images representing the pre-stage, middle stage, and post-stage of identifying vulnerable nodes in the interdependent

network [39]. In the case of the electricity network **(a)-(c)**, direct destruction is denoted by green markers, while yellow markers represent indirect destruction. For the road network **(d)-(f)**, light green indicates direct road collapse, magenta represents roads affected by ruined or non-functioning power stations, and red denotes roads separated from the largest connected component after the series of damages.

The impact of node selection in the electricity network is evidently more pronounced in terms of both the extent of influence and power decrease compared to the coupled network. For instance, damaging two nodes in the single electricity network reduces the power to approximately $6.5e9$, whereas in the coupled network, the power remains above $7e9$ even after the same number of nodes are damaged. However, when considering the connectivity in the road network, the topological changes undergo a significant shift as nodes in the electricity network are damaged. As the damage continues, the coverage area of normally operating power stations and the largest connected component become increasingly limited. This demonstrates the existence of cascade failure and validates our framework's ability to accurately characterize the vulnerability of urban infrastructures.

5 RELATED WORK

5.1 Data science in urban infrastructure

Urban infrastructure, including road networks, communication networks, electricity networks, etc., is one of the most important elements in the city, and there are many applications for data-driven urban computing research. The early efforts mainly focus on the modeling or prediction of urban infrastructure. For road networks, the existing works pay a lot of attention to the problem of road traffic forcasting [8], road traffic analaysis [25], travel time prediction [36], or using the road network to predict other elements in the city such as air quality [38]. That is, the existing works are still concentrated on one specific kind of infrastructure network, ignoring the joint modeling of the interdependence relations among different networks, which we aim to address in this work.

Despite the early efforts, the resilience and vulnerability of urban infrastructure are less explored in the literature. In this work, we approach the problem of finding the critical node in the interdependent infrastructure network, with extensive applications in the real world.

5.2 Data-driven optimization on graph

Graphs widely exist in plenty of domains, such as social networks, user-item graphs in recommender systems, road networks, etc. The ubiquity of graphs makes the optimization task of graphs an important problem in many real-world applications. The typical optimization problems include the traveling salesman problem (TSP) [20], maximum independent set (MIS) [34], maximum cut (MaxCut) [12], etc. However, despite their importance and wide applications, these optimization problems are always NP-hard, leading to high-cost time complexity that is exponentially related to the graph size. The traditional methods to address it are based on heuristics guided by real-world experience from human experts. However, although heuristic methods may not perform badly, they still suffer from limitations, which can be resolved by data-driven

(a) Pre-stage of Electricity Network   (b) Mid-stage of Electricity Network   (c) Post-stage of Electricity Network

(d) Pre-stage of Road Network   (e) Mid-stage of Road Network   (f) Post-stage of Road Network

Figure 5: Process of finding vulnerable nodes in interdependent network

machine learning models [2]. First, the heuristic methods highly rely on problem-specific designs, which will be challenging for new scenarios. Second, the data-driven methods can extract useful features or prediction signals that may be implicit and hard to explain, making them less likely to be proposed by human experts. Khalil *et al.* [21] proposed to learn graph states with graph neural networks and greedily generate actions until the criteria are met, supporting the data-driven solutions for Minimum Vertex Cover, Maximum Cut, and Traveling Salesman problems. Li *et al.* [24] propose to use a graph convolutional network to predict whether a given vertex belongs to the optimal set or not, converting the original problem to a supervised-learning task. The authors further combine the tree search method with the neural network module to ensure the exploration ability of the approach.

Different from the existing works which only consider small graphs, in this work, we approach a real-world problem in the urban infrastructure with far larger graph and heterogeneous relations, along with the complex environment.

6 CONCLUSION

Addressing the vulnerability of urban infrastructure interdependent network, we propose a data-driven framework using graph neural network and reinforcement learning to detect vulnerable nodes in urban infrastructure networks. Requiring no prior expert knowledge but just the connected relationships between different types of nodes in the real world, it trains on the complicated network quite efficiently. Thanks to the transferring ability and robustness of our framework, we can ignore some slight perturbations in our data and trust our model's performance in terms of effectiveness. Besides, our framework also illustrates the necessity of an explicit representation for the complicated network. Our framework captures the risk of cascade failure and discovers vulnerable nodes in real-world systems, which turns out to be dramatically useful in protecting fragile nodes in urban infrastructure and designing more robust topologies for cities.

ACKNOWLEDGMENTS

REFERENCES

[1] Bahrulla Abdulla and Bjorn Birgisson. 2020. Predicting road network vulnerability to fluvial flooding using machine learning classifiers: Case study of Houston during Hurricane Harvey. In *Construction Research Congress 2020: Computer Applications*. 38–47.

[2] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. 2021. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research* 290, 2 (2021), 405–421.

[3] Charles D Brummitt, Raissa M D'Souza, and Elizabeth A Leicht. 2012. Suppressing cascades of load in interdependent networks. *Proceedings of the national academy of sciences* 109, 12 (2012), E680–E689.

[4] Sergey V Buldyrev, Roni Parshani, Gerald Paul, H Eugene Stanley, and Shlomo Havlin. 2010. Catastrophic cascade of failures in interdependent networks. *Nature* 464, 7291 (2010), 1025–1028.

[5] Reuven Cohen, Keren Erez, Daniel Ben-Avraham, and Shlomo Havlin. 2000. Resilience of the internet to random breakdowns. *Physical review letters* 85, 21 (2000), 4626.

[6] Paul Collier and Anthony J Venables. 2016. Urban infrastructure for development. *Oxford Review of Economic Policy* 32, 3 (2016), 391–409.

[7] Huang Darong, Shen Libing, and Zhao Ling. 2015. Vulnerability analysis of urban road network based on complex network theory. *Journal of Chongqing Jiaotong University (Natural Science)* 34, 1 (2015), 110.

[8] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, Linhong Zhu, Rose Yu, and Yan Liu. 2016. Latent space model for road networks to predict time-varying traffic. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1525–1534.

[9] Sarah Dunn, Sean Wilkinson, and Alistair Ford. 2016. Spatial structure and evolution of infrastructure networks. *Sustainable cities and society* 27 (2016), 23–31.

[10] Irene Eusgeld, Wolfgang Kröger, Giovanni Sansavini, Markus Schläpfer, and Enrico Zio. 2009. The role of network theory and object-oriented modeling within a framework for the vulnerability analysis of critical infrastructures. *Reliability Engineering & System Safety* 94, 5 (2009), 954–963.

[11] Changjun Fan, Li Zeng, Yizhou Sun, and Yang-Yu Liu. 2020. Finding key players in complex networks through deep reinforcement learning. *Nature machine intelligence* 2, 6 (2020), 317–324.

[12] Paola Festa, Panos M Pardalos, Mauricio GC Resende, and Celso C Ribeiro. 2002. Randomized heuristics for the MAX-CUT problem. *Optimization methods and software* 17, 6 (2002), 1033–1058.

[13] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2023. A survey of graph neural networks for recommender systems: challenges, methods, and directions. *ACM Transactions on Recommender Systems* 1, 1 (2023), 1–51.

[14] Jianxi Gao, Baruch Barzel, and Albert-László Barabási. 2016. Universal resilience patterns in complex networks. *Nature* 530, 7590 (2016), 307–312.

[15] Marco Grassia, Manlio De Domenico, and Giuseppe Mangioni. 2021. Machine learning dismantling and early-warning signals of disintegration in complex systems. *Nature Communications* 12, 1 (2021), 5190.

[16] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[17] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[18] Paul Hines, Eduardo Cotilla-Sanchez, and Seth Blumsack. 2010. Do topological models provide good information about electricity infrastructure vulnerability? *Chaos: An Interdisciplinary Journal of Nonlinear Science* 20, 3 (2010), 033122.

[19] Swami Iyer, Timothy Killingback, Bala Sundaram, and Zhen Wang. 2013. Attack robustness and centrality of complex networks. *PloS one* 8, 4 (2013), e59613.

[20] Michael Jünger, Gerhard Reinelt, and Giovanni Rinaldi. 1995. The traveling salesman problem. *Handbooks in operations research and management science* 7 (1995), 225–330.

[21] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. 2017. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems* 30 (2017).

[22] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[23] Yusheng Li, Yilun Shang, and Yiting Yang. 2017. Clustering coefficients of large networks. *Information Sciences* 382 (2017), 350–358.

[24] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. 2018. Combinatorial optimization with graph convolutional networks and guided tree search. *Advances in neural information processing systems* 31 (2018).

[25] Wei Liu, Yu Zheng, Sanjay Chawla, Jing Yuan, and Xie Xing. 2011. Discovering spatio-temporal causal interactions in traffic data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1010–1018.

[26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[27] Flaviano Morone and Hernán A Makse. 2015. Influence maximization in complex networks through optimal percolation. *Nature* 524, 7563 (2015), 65–68.

[28] Hancheng Ren, Shu Liu, Min Li, Hongping Zhang, Huiying Wang, Xiaoli Hao, and Jie Cui. 2022. Topological Analysis and Application of Urban Drainage Network. *Water* 14, 22 (2022), 3732.

[29] Hai-Peng Ren, Jihong Song, Rong Yang, Murilo S Baptista, and Celso Grebogi. 2016. Cascade failure analysis of power grid using new load distribution law and node removal rule. *Physica A: Statistical Mechanics and its Applications* 442 (2016), 239–251.

[30] Saharon Rosset, Ji Zhu, and Trevor Hastie. 2003. Margin maximizing loss functions. *Advances in neural information processing systems* 16 (2003).

[31] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[32] Csaba Szepesvari. 2010. Algorithms for reinforcement learning: Synthesis lectures on artificial intelligence and machine learning. *Morgan and Claypool* (2010).

[33] Xianfeng Tang, Yandong Li, Yiwei Sun, Huaxiu Yao, Prasenjit Mitra, and Suhang Wang. 2020. Transferring robustness for graph neural network against poisoning attacks. In *Proceedings of the 13th international conference on web search and data mining*. 600–608.

[34] Robert Endre Tarjan and Anthony E Trojanowski. 1977. Finding a maximum independent set. *SIAM J. Comput.* 6, 3 (1977), 537–546.

[35] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. 2019. Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. *arXiv preprint arXiv:1909.01315* (2019).

[36] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 25–34.

[37] Aviv Yehezkel and Reuven Cohen. 2012. Degree-based attacks and defense strategies in complex networks. *Physical Review E* 86, 6 (2012), 066114.

[38] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 186–194.

[39] Jun Zhang, Depeng Jin, and Yong Li. 2022. Mirage: an efficient and extensible city simulation framework (systems paper). In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 1–4.

[40] Xiaojin Jerry Zhu. 2005. Semi-supervised learning literature survey. (2005).

A   APPENDIX

A.1   Setting for Transferablity Experiment

We create mask graphs to represent graphs with slight perturbations. We maintain the same number of nodes but allow for changes in edges, such as deletion and addition, in the electricity network, primary road network, and coupled network. We then retrain the new node embeddings using Algorithm 1 and feed them into the value network of our primary agent to calculate the corresponding metrics introduced in Section 4.1.2 during the node damaging process. Finally, we compare the results with the baseline, including CI and DE, on the newly generated topologies.

Algorithm 1 Retrain Progress for Mask Graph

Input: $\mathbf{F}_v = \{\mathbf{f}_v, \forall v \in \mathcal{V}\} \in \mathbb{R}^{d \times |\mathcal{V}|}$; $\mathcal{G}^{mask} = (\mathcal{V}, \mathcal{E})$;
Output: $\mathbf{F}_v^{new} \in \mathbb{R}^{d \times |\mathcal{V}|}$

1: **for** epoch = $1 \rightarrow M$ **do**
2:     embedding $\Leftarrow$ GNN($\mathbf{F}_v, \mathcal{G}^{mask}$)
3:     distant $\Leftarrow$ **dist**(embedding, $\mathbf{F}_v$)
4:     reconstruction $\Leftarrow$ GNN(embedding, $\mathcal{G}^{mask}$)
5:     loss $\Leftarrow$ reconstruction $+A\times$ distant
6:     loss $\Rightarrow$ backward
7: **end for**
8: **return** embedding $\Leftarrow$ GNN($\mathbf{F}_v, \mathcal{G}^{mask}$)