Interdependent Causal Networks for Root Cause Localization

Dongjie Wang[*,†]
University of Central Florida
FL, USA
dwang@nec-labs.com

Zhengzhang Chen[*,§]
NEC Laboratories America Inc.
NJ, USA
zchen@nec-labs.com

Jingchao Ni
AWS AI Labs, Amazon
WA, USA
jingchni@amazon.com

Tong Liang
NEC Laboratories America Inc.
NJ, USA
ltong@nec-labs.com

Zheng Wang
University of Utah
UT, USA
u1208847@utah.edu

Yanjie Fu
University of Central Florida
FL, USA
yanjie.fu@ucf.edu

Haifeng Chen
NEC Laboratories America Inc.
NJ, USA
haifeng@nec-labs.com

ABSTRACT

The goal of root cause analysis is to identify the underlying causes of system problems by discovering and analyzing the causal structure from system monitoring data. It is indispensable for maintaining the stability and robustness of large-scale complex systems. Existing methods mainly focus on the construction of a single effective isolated causal network, whereas many real-world systems are complex and exhibit interdependent structures (*i.e.*, multiple networks of a system are interconnected by cross-network links). In interdependent networks, the malfunctioning effects of problematic system entities can propagate to other networks or different levels of system entities. Consequently, ignoring the interdependency results in suboptimal root cause analysis outcomes.

In this paper, we propose REASON, a novel framework that enables the automatic discovery of both intra-level (*i.e.*, within-network) and inter-level (*i.e.*, across-network) causal relationships for root cause localization. REASON consists of Topological Causal Discovery (TCD) and Individual Causal Discovery (ICD). The TCD component aims to model the fault propagation in order to trace back to the root causes. To achieve this, we propose novel hierarchical graph neural networks to construct interdependent causal networks by modeling both intra-level and inter-level non-linear causal relations. Based on the learned interdependent causal networks, we then leverage random walk with restarts to model the network propagation of a system fault. The ICD component focuses on capturing abrupt change patterns of a single system entity. This component examines the temporal patterns of each entity's metric data (*i.e.*, time series), and estimates its likelihood of being a root cause based on the Extreme Value theory. Combining the topological and individual causal scores, the top K system entities are identified as root causes. Extensive experiments on three real-world datasets validate the effectiveness of the proposed framework.

CCS CONCEPTS

• **Computing methodologies** → **Causal reasoning and diagnostics**; *Unsupervised learning*.

KEYWORDS

Interdependent Networks; AIOps; Causal Structure Learning; Root Cause Analysis; Graph Neural Networks; Network Propagation

ACM Reference Format:
Dongjie Wang[*,†], Zhengzhang Chen[*,§], Jingchao Ni, Tong Liang, Zheng Wang, Yanjie Fu, and Haifeng Chen. 2023. Interdependent Causal Networks for Root Cause Localization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599849

1 INTRODUCTION

Root Cause Analysis (RCA) refers to the process of identifying the root causes of system faults using surveillance metrics data [4, 22]. It has been widely used in IT operations, industrial process control, telecommunications, *etc.*, because a failure or malfunction in these systems would drastically affect user experiences and result in financial losses. For instance, an intermittent outage of Amazon Web Services can result in a loss of around $210 millions [23]. To maintain the reliability and robustness of such systems, Key Performance Indicators (KPIs), such as latency or connection time in a microservice system, and metrics data, such as CPU/memory usages in a microservice system are often monitored and recorded in real-time for system diagnosis. The intricacy of these systems and the magnitude of the monitoring data, however, make manual

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599849

[*]The authors contribute equally.
†Work done during an internship at NEC Laboratories America.
§Corresponding author.

root cause analysis unacceptably expensive and error-prone. Consequently, an efficient and effective root cause analysis that enables rapid service recovery and loss mitigation is essential for the steady operation and robust management of large-scale complex systems.

Prior studies on root cause analysis [13, 25–27, 30] have mainly focused on a simplified scenario, where the target system is modeled as a single isolated causal graph, and the system's malfunctioning effects can only propagate within the same network of entities. For instance, to identify various types of service root causes, Liu *et al.* [27] generated a service call graph based on domain-specific software and rules. To discover the root causes of microservice system failures, Chen *et al.* [26] constructed a directed acyclic graph that depicts the invoking relations among microservice applications. These methods have been applied to some uncomplicated systems with isolated network structures.

However, a real-world complex system usually consists of multiple networks that coordinate in a highly complex manner [3, 12, 18]. These networks are interconnected, and if one network's system entity fails, it may spread to its dependent entities in other networks, which may then cause cascading damages or failures [10, 28] that could circulate throughout the interconnected levels with catastrophic consequences. For instance, Figure 1 shows the malfunction of the pod *Django-search* first spreads to the server network and causes the fault of the server *Compute-1*; Then, the malfunctioning effects spread to the pod network of *Compute-2* and causes the faults of the pod *Mongodb* and *Mysql*; Finally, the pod *Sdn* in *Infra-1* is also affected, resulting in the system failure. In this failure case, it is quite difficult to pinpoint the root cause *Django-search* if we only model the server network (*i.e.*, the three servers) or one of the three pod networks of the microservice system. Thus, modeling the interconnected multi-network structures is vital for comprehensive understanding of the complex system and effective root cause localization.

Recently, a promising approach for modeling such interconnected structures in complex systems has emerged through the concept of interdependent networks (or network of networks) [2, 3, 28, 32]. In interdependent networks, each node of the main network can be represented as a domain-specific network. Let us elaborate using the example in Figure 1 again. Here, the dashed network represents a server/machine network (the main network), where the nodes are three different servers and edges/links indicate the causal relations among different servers. Each node of this main network is further represented as a pod network (the domain-specific network), where nodes are pods and edges denote their causal relations. Collectively, we call this structure a (server-pod) interdependent networks. And since all the edges in these interdependent networks indicate causal dependencies, we further call it *interdependent causal networks*. Interdependent networks have been widely used in the study of various topics, including the academic influence of scholars [34], the spreading pattern of rumors in the complex social network [32], and etc. However, existing methods only consider physical or statistical correlations, but not causation, and thus cannot be directly applied for locating root causes.

Enlightened by the interdependent networks, this paper aims to learn interdependent causal relationships from monitoring metrics in multi-network systems for accurately identifying root causes when a system failure/fault occurs. Formally, given the system

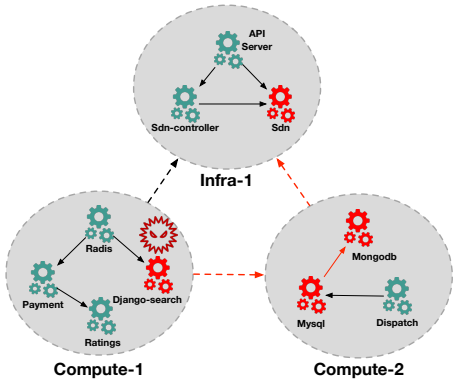


Figure 1: An example of interdependent causal networks. The main network is represented (*i.e.*, server network) by dashed nodes and edges. The domain-specific networks (*i.e.*, pod networks) are represented by solid nodes and edges. The edges highlighted by red colors indicate how the malfunctioning effects of a root cause propagate in the system.

KPI data, the multi-level interconnected system entities, and their metrics data (*i.e.*, time series), our goal is to learn interdependent causal structures for discovering the root causes of system failures. There are two major challenges in this task:

- **Challenge 1: Learning interdependent causal networks and modeling fault propagation in interdependent causal networks.** As aforementioned, in real-world systems with interdependent networks structures, malfunctioning effects of root causes can propagate to other nodes of the same level or different levels (*i.e.*, main network level and domain-specific network level), resulting in catastrophic failure of the entire system. To capture such propagation patterns for root cause localization, we need to learn the causal relationships not only within the same level but also across levels. After modeling the interdependent causal relationships, we still need to model the propagation of malfunctioning effects in the learned causal interdependent networks.
- **Challenge 2: Identifying abrupt change patterns from the metrics data of an individual system entity.** Besides the topological patterns, metrics data associated with the system entities can exhibit abrupt change patterns during the incidence of system faults, particularly those that are short-lived (*e.g.*, fail-stop failures). The malfunctioning effects of the root cause may end quickly before they can spread. Thus, the temporal patterns from the metrics data can provide individual causal insights for locating root causes. The challenge is how to capture abrupt change patterns and determine the individual causal effect associated with the system failure.

To address these challenges, in this paper, we propose REASON, a generic interdependent causal networks based framework, for root cause localization in complex systems with interdependent network structures. REASON consists of Topological Causal Discovery (TCD) and Individual Causal Discovery (ICD). For the TCD component, the assumption is that the malfunctioning effects of root causes can propagate to other system entities of the same level or different levels over time [10, 28]. To capture such propagation patterns, we propose a hierarchical graph neural networks based

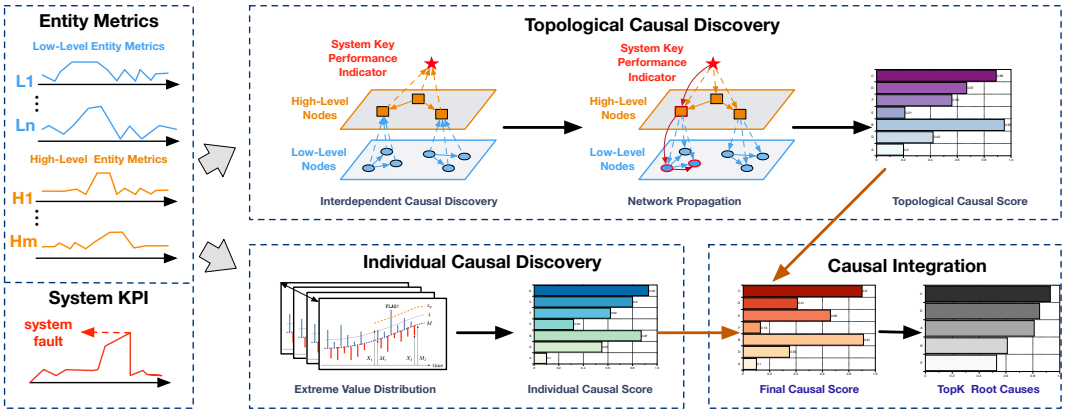


Figure 2: The overview of the proposed framework REASON. It consists of three major steps: topological causal discovery, individual causal discovery, and causal integration.

causal discovery method to discover both intra-level (*i.e.*, within-network) and inter-level (*i.e.*, across-network) non-linear causal relationships. Then, we leverage a random walk with restarts to model the network propagation of a system fault. The ICD component, on the other hand, focuses on individual causal effects, by analyzing the metrics data (*i.e.*, time series) of each system entity. Especially considering the short-lived failure cases (*e.g.*, fail-stop failures), there may be no propagation patterns. We design an Extreme Value theory based method to capture the abrupt fluctuation patterns and estimate the likelihood of each entity being a root cause. Finally, we integrate the findings of TCD and ICD, and output the system entities with the top-K greatest causal scores as the root causes. Extensive experiments and case studies are conducted on three real-world datasets to validate the efficacy of our work.

2 PRELIMINARIES

System Key Performance Indicator (KPI). A KPI is a monitoring time series that indicates the system status. For instance, in a microservice system, the latency or connection time can be used to assess the system status. The smaller the latency, the higher the system's performance quality. If the connection time is too long, it is likely that the system has failed.

Entity Metrics. Entity metrics data can be collected by monitoring different levels of system entities. It usually contains a number of metrics, which indicate the status of a system's underlying entity. For example, in a microservice system, the underlying entity can be a physical machine, container, virtual machine, pod, etc. And the corresponding metrics can be CPU utilization, memory utilization, disk IO utilization, etc. The data for all these metrics are essentially time series. An anomalous metric of a microservice's underlying entity can be the potential root cause of an anomalous system latency/connection time, which indicates a microservice failure.

Interdependent Networks (INs). Interdependent networks model the interconnections of multiplex networks [34, 35]. Given a $g \times g$ main network $\mathbf{G}$, a set of domain-specific networks $\mathcal{A} = \{\mathbf{A}_1, \cdots, \mathbf{A}_g\}$, and an edge set $\mathbf{E}$ that represents the edges between the nodes in $\mathcal{A}$ and the nodes in $\mathbf{G}$, INs are defined as a triplet $\mathcal{R} =< \mathbf{G}, \mathcal{A}, \mathbf{E} >$. The node set in $\mathbf{G}$, *a.k.a.* high-level nodes, is denoted by $\mathbf{V}^G$, and the node set in $\mathcal{A}$, *a.k.a.* low-level nodes, is denoted by $\mathbf{V}^{\mathcal{A}} = (\mathbf{V}^{A_1}, \cdots, \mathbf{V}^{A_g})$.

As a special type of INs, interdependent causal networks represent the INs with the edges indicating causal relations.

Take Figure 1 as an example, the dashed network is the main network $\mathbf{G}$, which has three server nodes, including *Compute-1*, *Compute-2*, and *Infra-1*. Each of these main nodes contains a domain-specific network that is made up of several applications/pods. For instance, the main node *Compute-2* is further represented as a domain-specific network with three pod nodes, including *Mysql*, *Mongodb*, and *Dispatch*. And the solid edges indicate the causal relationships between different pods, while the dashed edges indicate the causal relationships between different servers.

Without loss of generality, we focus on two levels of system entities. Our goal is to identify root causes by automatically learning interdependent causal relations between different levels of system entities and the system KPI. The identified root causes are low-level system entities to reflect fine-grained root cause detection.

Problem Statement. Given metrics/sensor data of multi-level system entities corresponding to high-level and low-level nodes in main and domain-specific networks $\{\mathbf{X}^G, \mathbf{X}^{\mathcal{A}}\}$, and system key performance indicator $\mathbf{y}$, the problem is to construct an interdependent causal network $\mathcal{R} =< \mathbf{G}, \mathcal{A}, \mathbf{E} >$, and identify the top K low-level nodes in $\mathbf{V}^{\mathcal{A}}$ that are most relevant to $\mathbf{y}$.

3 METHODOLOGY

We present REASON, an interdependent causal network based framework for root cause localization. As illustrated in Figure 2, REASON includes three major steps: 1) topological causal discovery; 2) individual causal discovery; and 3) causal integration.

3.1 Topological Causal Discovery

Root causes (*i.e.,* the system entities that cause the system failures or faults) could propagate malfunctioning or fault effects to other system entities of the same network or across different networks over time [10, 15, 28], which makes real root causes hard to locate. To address this challenge, we propose a hierarchical graph neural network based causal discovery method to construct interdependent causal graphs among low-level and high-level system entities. Failure propagation is modeled on the learned causal structures to provide topological guidance for locating root causes by simulating the malfunctioning effects of root causes.

3.1.1 Hierarchical Graph Neural Network based Interdependent Causal Discovery. There can be more than one entity metric (*i.e.*, multi-variate time series) per system entity (refer to Section 2). For each individual metric, we learn interdependent causal graphs among different system entities using the same learning strategy. To ease the description, we take one metric of system entities as an example to illustrate the interdependent causal discovery process.

The metric of system entities (*i.e.*, high-level or low-level) is a multivariate time series $\{\mathbf{x}_0, \cdots, \mathbf{x}_T\}$. The metric value at the t-th time step is $\mathbf{x}_t \in \mathbb{R}^d$, where d is the number of entities. The data can be modeled using the VAR model [45, 48], whose formulation is given by:

$$\mathbf{x}_t^\top = \mathbf{x}_{t-1}^\top \mathbf{B}_1 + \cdots + \mathbf{x}_{t-p}^\top \mathbf{B}_p + \boldsymbol{\epsilon}_t^\top, \quad t = \{p, \cdots, T\} \tag{1}$$

where p is the time-lagged order, $\boldsymbol{\epsilon}_t$ is the vector of error variables that are expected to be non-Gaussian and independent in the temporal dimension, $\{\mathbf{B}_1, \cdots, \mathbf{B}_p\}$ are the weighted matrix of time-lagged data. In the VAR model, the time series at t, $\mathbf{x}_t$, is assumed to be a linear combination of the past p lags of the series.

Assuming that $\{\mathbf{B}_1, \cdots, \mathbf{B}_p\}$ is constant across time, the Equation (1) can be extended into a matrix form:

$$\mathbf{X} = \tilde{\mathbf{X}}_1 \mathbf{B}_1 + \cdots + \tilde{\mathbf{X}}_p \mathbf{B}_p + \boldsymbol{\varepsilon} \tag{2}$$

where $\mathbf{X} \in \mathbb{R}^{m \times d}$ is a matrix and its each row is $\mathbf{x}_t^\top$; $\{\tilde{\mathbf{X}}_1, \cdots, \tilde{\mathbf{X}}_p\}$ are the time-lagged data.

To simplify Equation 2, let $\tilde{\mathbf{X}} = [\tilde{\mathbf{X}}_1 | \cdots | \tilde{\mathbf{X}}_p]$ with its shape of $\mathbb{R}^{m \times pd}$ and $\mathbf{B} = [\mathbf{B}_1 | \cdots | \mathbf{B}_p]$ with its shape of $\mathbb{R}^{m \times pd}$. Here, $m = T + 1 - p$ is the effective sample size, because the first p elements in the metric data have no sufficient time-lagged data to fit Equation 2. After that, we apply the QR decomposition to the weight matrix $\mathbf{B}$ to transform Equation 2 as follows:

$$\mathbf{X} = \tilde{\mathbf{X}} \hat{\mathbf{B}} \mathbf{W} + \boldsymbol{\varepsilon} \tag{3}$$

where $\hat{\mathbf{B}} \in \mathbb{R}^{m \times pd}$ is the weight matrix of time-lagged data in the temporal dimension; $\mathbf{W} \in \mathbb{R}^{d \times d}$ is the weighted adjacency matrix, which reflects the relations among system entities.

A nonlinear autoregressive model allows $\mathbf{x}_t$ to evolve according to more general nonlinear dynamics [8]. In a forecasting setting, one promising way is to jointly model the nonlinear functions using neural networks [8, 24]. By applying neural networks f to Equation 3, we have

$$\mathbf{X} = f(\tilde{\mathbf{X}} \hat{\mathbf{B}} \mathbf{W}; \boldsymbol{\Theta}) + \boldsymbol{\varepsilon} \tag{4}$$

where $\boldsymbol{\Theta}$ is the set of parameters of f.

Given the data $\mathbf{X}$ and $\tilde{\mathbf{X}}$, here our goal is to estimate weighted adjacency matrices $\mathbf{W}$ that correspond to directed acyclic graphs (DAGs). The causal edges in $\mathbf{W}$ go only forward in time, and thus they do not create cycles. In order to ensure that the whole network is acyclic, it thus suffices to require that $\mathbf{W}$ is acyclic. Minimizing the least-squares loss with the acyclicity constraint gives the following optimization problem:

$$\min \frac{1}{m} \left\| \mathbf{X} - f(\tilde{\mathbf{X}} \hat{\mathbf{B}} \mathbf{W}; \boldsymbol{\Theta}) \right\|^2 \quad s.t.\ \mathbf{W} \text{ is acyclic}, \tag{5}$$

To learn $\mathbf{W}$ in an adaptive manner, we adopt the following layer:

$$\mathbf{W} = \text{RELU}(\tanh(\mathbf{W}_+ \mathbf{W}_-^\top - \mathbf{W}_- \mathbf{W}_+^\top)), \tag{6}$$

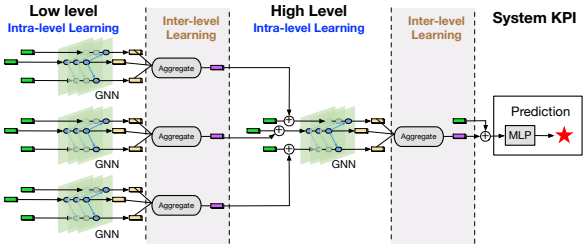


Figure 3: The learning process of hierarchical GNNs. Intra-level learning captures causation within the same-level system entities. Inter-level learning aggregates low-level information to high-level for constructing cross-level causation.

where $\mathbf{W}_+ \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_- \in \mathbb{R}^{d \times d}$ are two parameter matrices. This learning layer aims to enforce the asymmetry of $\mathbf{W}$, because the propagation of malfunctioning effects is unidirectional and acyclic from root causes to subsequent entities. In the following sections, $\mathbf{W}^G$ denotes the causal relations between high-level nodes and $\mathbf{W}^{\mathcal{A}}$ denotes the causal relations between low-level nodes.

Then, the causal structure learning for the interdependent networks can be divided into intra-level learning and inter-level learning. Intra-level learning is to learn the causation among the same level of nodes, while inter-level learning is to learn the cross-level causation. To model the influence of low-level nodes on high-level nodes, we aggregate low-level information into high-level nodes in inter-level learning. Figure 3 shows the learning process.

For **intra-level learning**, we adopt the same learning strategy to learn causal relations among both high-level nodes and low-level nodes. Specifically, we first apply L layers of GNN to the time-lagged data $\{\mathbf{x}_{t-1}, \cdots, \mathbf{x}_{t-p}\} \in \mathbb{R}^{d \times p}$ to obtain its embedding. In the l-th layer, the embedding $\mathbf{z}^{(l)}$ is obtained by aggregating the nodes' embedding and their neighbors' information at the $l-1$ layer. Then, the embedding at the last layer $\mathbf{z}^{(L)}$ is used to predict the metric value at the time step t by an MLP layer. This process can be represented as

$$\begin{cases} \mathbf{z}^{(0)} = [\mathbf{x}_{t-1}, \cdots, \mathbf{x}_{t-p}], \\ \mathbf{z}^{(l)} = \text{GNN}(\text{Cat}(\mathbf{z}^{(l-1)}, \mathbf{W} \cdot \mathbf{z}^{(l-1)}) \cdot \mathbf{B}^{(l)}), \\ \check{\mathbf{x}}_t = \text{MLP}(\mathbf{z}^{(L)}; \boldsymbol{\Theta}), \end{cases} \tag{7}$$

where Cat is the concatenation operation; $\mathbf{B}^{(l)}$ is the weight matrix of the l-th layer; GNN is activated by the RELU function to capture non-linear correlations in the time-lagged data. Our goal is to minimize the difference between the actual value $\mathbf{x}_t$ and the predicted value $\check{\mathbf{x}}_t$. Thus, the optimization objective is defined as follows

$$\mathcal{L} = \frac{1}{m} \sum_t (\mathbf{x}_t - \check{\mathbf{x}}_t)^2 \tag{8}$$

As shown in Figure 3, we conduct intra-level learning for the low-level and high-level system entities for constructing $\mathbf{W}^{\mathcal{A}}$ and $\mathbf{W}^G$, respectively. The optimization objectives for the low-level and high-level causal relations, in the same format as Equation 8, are denoted by $\mathcal{L}_{\mathcal{A}}$ and $\mathcal{L}_G$, respectively.

For **inter-level learning**, we aggregate the information of low-level nodes to the high-level nodes for constructing the cross-level causation. So, the initial embedding of high-level nodes $\mathbf{z}^{(0)}$ is the

concatenation of their time-lagged data $\{\mathbf{x}_{t-1}, \cdots, \mathbf{x}_{t-p}\}$ and aggregated low-level embeddings, which can be formulated as follows

$$\mathbf{z}^{(0)} = \text{Cat}([\mathbf{x}_{t-1}, \cdots, \mathbf{x}_{t-p}], \mathbf{W} \cdot \mathbf{z}^{(L)}) \tag{9}$$

where $\mathbf{W}$ is a weight matrix that controls the contributions of low-level embeddings to high-level embeddings. As shown in Figure 3, there are two inter-level learning parts. The first one is used to learn the cross-level causal relations between low-level and high-level nodes, denoted by $\mathbf{W}^{\mathcal{A}G}$. The second one is used to construct the causal linkages between high-level nodes and the system KPI, denoted by $\mathbf{W}^{GS}$. During this process, we predict the value of the system KPI at the time step t and aim to make the predicted values close to the actual ones. Hence, we formulate the optimization objective $\mathcal{L}_S$, whose format is the same as Equation 8.

In addition, the learned interdependent causal graphs must meet the acyclicity requirement. Since the cross-level causal relations $\mathbf{W}^{\mathcal{A}G}$ and $\mathbf{W}^{GS}$ are unidirectional, only $\mathbf{W}^{\mathcal{A}}$ and $\mathbf{W}^{G}$ need to be acyclic. To achieve this goal, inspired by the work [50], we use the trace exponential function: $h(\mathbf{W}) = tr(e^{\mathbf{W}\circ\mathbf{W}}) - d = 0$ that satisfies $h(\mathbf{W}) = 0$ if and only if $\mathbf{W}$ is acyclic. Here, $\circ$ is the Hadamard product of two matrices. Meanwhile, to enforce the sparsity of $\mathbf{W}^{\mathcal{A}}$, $\mathbf{W}^{G}$, $\mathbf{W}^{\mathcal{A}G}$, and $\mathbf{W}^{GS}$ for producing robust causation, we use the $L1$-norm to regularize them. So, the final optimization objective is

$$\begin{aligned}\mathcal{L}_{final} = &\ (\mathcal{L}_{\mathcal{A}} + \mathcal{L}_G + \mathcal{L}_S) \\ &+ \lambda_1(\left\|\mathbf{W}^{\mathcal{A}}\right\|_1 + \left\|\mathbf{W}^{G}\right\|_1 + \left\|\mathbf{W}^{\mathcal{A}G}\right\|_1 + \left\|\mathbf{W}^{GS}\right\|_1) \\ &+ \lambda_2(h(\mathbf{W}^{\mathcal{A}}) + h(\mathbf{W}^{G}))\end{aligned} \tag{10}$$

where $\|\cdot\|_1$ is the element-wise $L1$-norm; λ_1 and λ_2 are two parameters that control the contribution of regularization items. We aim to minimize $\mathcal{L}_{final}$ through the L-BFGS-B solver. When the model converges, we construct interdependent causal networks through $\mathbf{W}^{\mathcal{A}}$, $\mathbf{W}^{G}$, $\mathbf{W}^{\mathcal{A}G}$, and $\mathbf{W}^{GS}$.

3.1.2 Network Propagation on Interdependent Causal Graphs. As aforementioned, starting from the root cause entity, malfunctioning effects will propagate to neighboring entities [13], and different types of system faults can trigger diverse propagation patterns. This observation motivates us to apply network propagation to the learned causal structure to mine the hidden actual root causes.

The learned interdependent causal structure is a directed acyclic graph, which reflects the causal relations from the low-level to the high-level to the system level. In order to trace back the root causes, we need to conduct a reverse analysis process. Thus, we transpose the learned causal structure to get $<< \mathbf{G}^\top, \mathcal{A}^\top, \mathbf{E} >, \text{KPI} >$[1], then apply a random walk with restart on the interdependent causal networks to estimate the topological causal score of each entity.

Specifically, the transition probabilities of a particle on the transposed structure can be denoted by

$$H = \begin{bmatrix} H_{\mathbf{GG}} & H_{\mathbf{G}\mathcal{A}} \\ H_{\mathcal{A}\mathbf{G}} & H_{\mathcal{A}\mathcal{A}} \end{bmatrix} \tag{11}$$

where $H_{\mathbf{GG}}$ and $H_{\mathcal{A}\mathcal{A}}$ depict the walks within the same-level network. $H_{\mathbf{G}\mathcal{A}}$ and $H_{\mathcal{A}\mathbf{G}}$ describe the walks across different level networks. Imagine that from the KPI node, a particle begins to visit the networks. The particle randomly selects a high-level or low-level node to visit, then the particle either jumps to the low-level nodes or walks in the current graph with a probability value $\Phi \in [0, 1]$. The higher the value of Φ is, the more possible the jumping behavior occurs. In detail, if a particle is located at a high-level node i in $\mathbf{G}$, the probability of the particle moving to the high-level node j is

$$H_{\mathbf{GG}}(i, j) = (1 - \Phi)\mathbf{G}^\top(i, j) / \sum_{k=1}^{g} \mathbf{G}^\top(i, k) \tag{12}$$

or jumping to the low-level node b with a probability

$$H_{\mathbf{G}\mathcal{A}}(i, b) = \Phi\mathbf{W}(i, b) / \sum_{k=1}^{gd} \mathbf{W}(i, k) \tag{13}$$

We apply the same strategy when the particle is located at a low-level node. The particle walking between different low-level nodes has a visiting probability of $H_{\mathcal{A}\mathcal{A}}$, whose calculation equation is similar to $H_{\mathbf{GG}}$. Moreover, the visiting probability from a low-level node to a high-level node is $H_{\mathcal{A}\mathbf{G}}$, whose calculation equation is similar to $H_{\mathbf{G}\mathcal{A}}$. The probability transition evolving equation of the random walk with restart can be formulated as

$$\tilde{\mathbf{p}}_{t+1}^\top = (1 - \varphi)\tilde{\mathbf{p}}_t^\top + \varphi\tilde{\mathbf{p}}_{rs}^\top \tag{14}$$

where $\tilde{\mathbf{p}}_{t+1}^\top \in \mathbb{R}^{g+gd}$ and $\tilde{\mathbf{p}}_t^\top \in \mathbb{R}^{g+gd}$ are the visiting probability distribution at different time steps; $\tilde{\mathbf{p}}_{rs}^\top \in \mathbb{R}^{g+gd}$ is the initial visiting probability distribution that depicts the visiting possibility of high-level or low-level nodes at the initialization step. $\varphi \in [0, 1]$ is the restart probability. When the visiting probability distribution is convergence, we regard the probability score of the low-level nodes as the associated topological causal score.

3.2 Individual Causal Discovery

In addition to the topological causal effects, the entity metrics of root causes themselves could fluctuate stronger than those of other system entities during the incidence of some system faults. And for some short-lived failure cases (*e.g.*, fail-stop failure), there may even be no propagation patterns. Thus, we propose to individually analyze such temporal patterns in order to provide individual causal guidance for locating root causes.

Compared with the values of entity metrics in normal time, the fluctuating values are extreme and infrequent. Inspired by [41], such extreme value follows the extreme value distribution, which is defined as:

$$U_\zeta : x \to exp(-(1 + \zeta x)^{-\frac{1}{\zeta}}), \quad \zeta \in \mathbb{R}, \quad 1 + \zeta x > 0. \tag{15}$$

where x is the original value and ζ is the extreme value index depending on the distribution of x. Let the probability of potential extreme value in x be q, the boundary[2] ϱ of normal value can be calculated through $\mathbb{P}(X > \varrho) = q$ based on U_ζ. However, since the distribution of x is unknown, ζ should be estimated. The Pickands-Belkema-de Haan theorem [38] provides an approach to estimate ζ, which is defined as follows:

Theorem 3.1. *The extrema of a cumulative distribution F converge to the distribution of U_ζ, denoted as $F \in D_\zeta$, if and only if a function*

[1]Here, $\mathbf{E}$ contains not only the edges between the nodes in $\mathcal{A}$ and the nodes in $\mathbf{G}$ but also the edges between the nodes in $\mathbf{G}$ and the node of system KPI.

[2]The boundary can be upper bound or lower bound of normal values.

δ exists, for all $x \in \mathbb{R}$ s.t. $1+\zeta x > 0$:

$$\frac{\overline{F}(\eta+\delta(\eta)x)}{\overline{F}(\eta)} \quad \overrightarrow{\eta \to \tau} \quad (1+\zeta x)^{-\frac{1}{\zeta}}, \tag{16}$$

where U_ζ refers to the extreme value distribution; D_ζ refers to a Generalized Pareto Distribution; η is a threshold for peak normal value; and τ is the boundary of the initial distribution. Assuming that η is a threshold for peak normal value, $X-\eta$ follows a Generalized Pareto Distribution (GPD) with parameters ζ and δ according to the theorem, which is defined as:

$$\mathbb{P}(X-\eta > x | X > \eta) \sim (1+\frac{\zeta x}{\delta(\eta)})^{-\frac{1}{\zeta}}. \tag{17}$$

We can utilize the maximum likelihood estimation method [6] to estimate ζ and δ. Then, the boundary value ϱ can be calculated by

$$\varrho \simeq \eta + \frac{\delta}{\zeta}((\frac{qn}{N_\eta})^{-\zeta} - 1). \tag{18}$$

where η, q can be provided by domain knowledge, n is the total number of observations, and N_η is the number of peak values (*i.e.*, the number of $X > \eta$).

Individual causal discovery is devised based on Equation (18). Specifically, we divide the metric data of one system entity into two segments. The first segment is used for initialization, and the second one is used for detection. For initialization, we first provide the probability of the extreme value q and the threshold of the peak value η using a mean excess plot-based method [6]. Then, we use the first time segment to estimate the boundary ϱ of normal value according to Equation (18). Here, η should be lower than ϱ. For detection, we compare each value in the second time segment with ϱ and η. If the value is larger than ϱ, the value is abnormal, so we store it. If the value is less than ϱ but larger than η, which means the boundary ϱ has been changed. Hence, we add it to the first segment and re-evaluate the parameters ζ and δ to get new boundaries. If the value is less than η, it is normal, so we ignore it. Finally, we can collect all abnormal values and normalize them using the Sigmoid function. The mean of the normalized values is the individual causal score of the associated system entity.

3.3 Causal Integration

Finally, we integrate the individual and topological causal scores of low-level system entities through the integration parameter $0 \leq \gamma \leq 1$, which can be represented as $\mathbf{q}_{final} = \gamma \mathbf{q}_{indiv} + (1-\gamma)\mathbf{q}_{topol}$. After that, we rank low-level nodes using $\mathbf{q}_{final}$ and select the top K results as the final root causes.

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Datasets. We evaluated REASON on the following three real-world datasets for the task of root cause localization. **1) AIOps**: This dataset was collected from a real micro-service system. This system has 234 microservice pods/applications (low-level system entities) that are deployed to 5 cloud servers (high-level system entities). The operators collected metrics data (*e.g.*, CPU Usage, Memory Usage) of high-level and low-level system entities from May 2021 to December 2021. There are 5 system faults during this time period. **2) WADI [1]**: This dataset was collected from a water distribution testbed, which owns 3 stages (high-level entities) and 123 sensors (low-level entities). It has 15 system faults collected in 16 days. In these datasets, low-level entities affiliate with high-level entities, and same-level entities invoke each other. **3) SWaT [29]**: This dataset was collected from a water treatment testbed, which consists of 6 stages (high-level entities) that have 51 sensors (low-level entities). It has 16 system faults collected in 11 days.

4.1.2 Evaluation Metrics. We evaluated the model performance with the following three widely-used metrics [27, 30]:
Precision@K (PR@K). It denotes the probability that the top-K predicted root causes are real, defined as

$$\text{PR@K} = \frac{1}{|\mathbb{A}|}\sum_{a\in\mathbb{A}}\frac{\sum_{i<K} R_a(i) \in V_a}{min(K, |V_a|)}, \tag{19}$$

where $\mathbb{A}$ is the set of system faults; a is one fault in $\mathbb{A}$; V_a is the real root causes of a; R_a is the predicted root causes of a; and i refers to the i-th predicted cause of R_a.
Mean Average Precision@K (MAP@K). It assesses the model performance in the top-K predicted causes from the overall perspective, defined as

$$\text{MAP@K} = \frac{1}{K|\mathbb{A}|}\sum_{a\in\mathbb{A}}\sum_{1\leq j\leq K}\text{PR@j}, \tag{20}$$

where a higher value indicates better performance.
Mean Reciprocal Rank (MRR). This metric measures the ranking capability of models. The larger the MRR value is, the further ahead the predicted positions of the root causes are; thus, operators can find the real root causes more easily. MRR is defined as

$$\text{MRR} = \frac{1}{\mathbb{A}}\sum_{a\in\mathbb{A}}\frac{1}{rank_{R_a}}, \tag{21}$$

where $rank_{R_a}$ is the rank number of the first correctly predicted root cause for system fault a.

4.1.3 Baselines. We compared REASON with the following five causal discovery models: **1) PC**[44] is a classic constraint-based method. It first identifies the skeleton of the causal graph with the independence test, then generates the orientation direction using the v-structure and acyclicity constraints. **2) C-LSTM**[48] captures the nonlinear Granger causality that existed in multivariate time series by using LSTM neural networks. **3) Dynotears**[36] is a score-based method that uses the structural vector autoregression model to construct dynamic Bayesian networks. **4) GOLEM**[33] employs a likelihood-based score function to relax the hard DAG constraint in NOTEARS. **5) GNN** is a simplified version of our causal discovery method. It only uses GNN to learn causal structures among low-level system entities.

Since none of the above baselines can be directly applied to learn the hierarchical interdependent causation, we only utilized the entity metrics to construct causation between low-level system entities and the system KPI. We then selected the top-K entities with the highest causal scores as the root causes. To verify the effectiveness of the network propagation module (see Section 3.1.2), we applied it to the causal structures learned by these baselines and analyzed model performance changes.

In addition, to study the impact of each technical component of REASON, we developed the following model variants: (1) To assess

Table 1: Overall performance *w.r.t.* SWaT dataset.

	PR@1	PR@3	PR@5	PR@7	PR@10	MAP@3	MAP@5	MAP@7	MAP@10	MRR
REASON	**25.0%**	**28.13%**	**66.67%**	**76.04%**	**84.38%**	**23.96%**	**35.0%**	**46.73%**	**57.60%**	**40.99%**
GNN	18.75%	19.79%	43.75%	52.08%	62.50%	18.06%	27.92%	33.63%	41.88%	34.77%
PC	12.5%	13.54%	34.38%	47.92%	58.33%	12.85%	20.42%	26.64%	35.0%	26.16%
C-LSTM	12.5%	13.54%	28.13%	40.63%	52.08%	13.89%	17.71%	23.81%	31.88%	29.35%
Dynotears	12.5%	29.17%	32.29%	34.38%	42.71%	20.14%	24.38%	26.93%	30.83%	27.85%
GOLEM	6.25%	7.29%	12.5%	39.58%	47.92%	7.64%	9.58%	16.96%	25.0%	22.36%

Table 2: Overall performance *w.r.t.* WADI dataset.

	PR@1	PR@3	PR@5	PR@7	PR@10	MAP@3	MAP@5	MAP@7	MAP@10	MRR
REASON	**28.57%**	**59.52%**	**65.0%**	**76.19%**	**79.76%**	**42.46%**	**50.62%**	**57.41%**	**63.76%**	**53.35%**
GNN	14.28%	26.19%	34.28%	42.86%	54.76%	21.83%	25.31%	30.15%	37.54%	32.71%
PC	7.14%	27.38%	35.0%	44.05%	50.0%	16.27%	23.90%	28.47%	34.57%	27.74%
C-LSTM	0%	20.24%	35.0%	47.62%	51.19%	11.51%	18.55%	25.83%	32.73%	24.40%
Dynotears	7.14%	14.29%	30.00%	29.76%	47.62%	10.71%	17.43%	20.95%	26.81%	22.23%
GOLEM	0%	19.05%	40.0%	46.43%	53.57%	9.92%	20.38%	27.82%	34.83%	23.48%

Table 3: Overall performance *w.r.t.* AIOps dataset.

	PR@1	PR@3	PR@5	PR@7	PR@10	MAP@3	MAP@5	MAP@7	MAP@10	MRR
REASON	**80.0%**	**80.0%**	**100.0%**	**100.0%**	**100.0%**	**80.0%**	**84.0%**	**88.57%**	**92.0%**	**84.0%**
GNN	20.0%	40.0%	40.0%	40.0%	60.0%	26.67%	32.0%	34.29%	38.0%	30.65%
PC	0%	20.0%	20.0%	40.0%	40.0%	13.33%	16.0%	22.86%	28.0%	14.0%
C-LSTM	0%	20.0%	20.0%	20.0%	20.0%	13.33%	16.0%	17.14%	18.0%	10.82%
Dynotears	20.0%	40.0%	40.0%	40.0%	40.0%	33.33%	36.0%	37.14%	38.0%	30.79%
GOLEM	20.0%	40.0%	40.0%	40.0%	40.0%	33.33%	36.0%	37.14%	38.0%	31.22%

Table 4: The influence of network propagation in terms of MAP@10

	PC		GLOEM		Dynotears		C-LSTM		GNN	
	Original	Propagate	Original	Propagate	Original	Propagate	Original	Propagate	Original	Propagate
SWaT	35.0%	**37.39%**	25.0%	**33.44%**	30.83%	**37.08%**	31.87%	**34.16%**	41.87%	**49.16%**
WADI	34.57%	**35.71%**	34.83%	**38.05%**	26.81%	**33.76%**	32.72%	**42.61%**	37.53%	**45.98%**
AIOPS	28.0%	**30.0%**	38.0%	**54.0%**	38.0%	**58.0%**	18.0%	**48.0%**	38.0%	**60.0%**

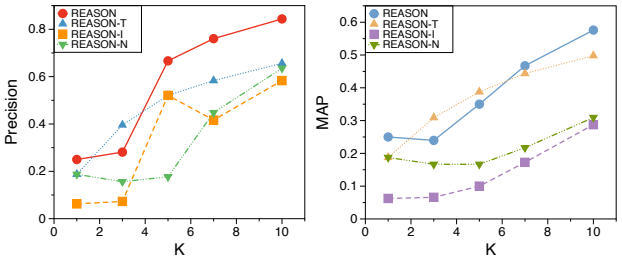
Table 5: The influence of network propagation in terms of MRR

	PC		GLOEM		Dynotears		C-LSTM		GNN	
	Original	Propagate	Original	Propagate	Original	Propagate	Original	Propagate	Original	Propagate
SWaT	26.16%	**32.27%**	22.36%	**30.42%**	27.85%	**33.98%**	29.35%	**32.85%**	34.77%	**40.43%**
WADI	27.74%	**30.74%**	23.48%	**25.89%**	22.22%	**34.28%**	24.39%	**33.27%**	32.71%	**36.40%**
AIOPS	14.0%	**25.35%**	31.22%	**37.74%**	30.79%	**50.77%**	10.82%	**24.73%**	30.65%	**62.48%**

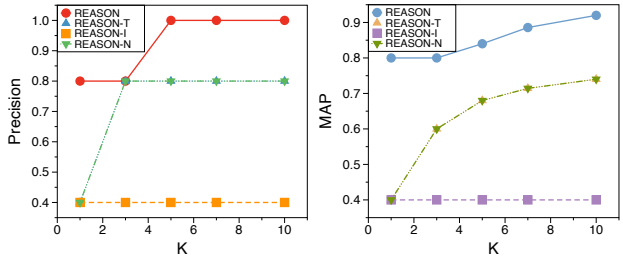
the benefits of inter-level learning (see Section 3.1), we implemented REASON-N by removing the inter-level learning in topological causal discovery while keeping the intra-level learning of low-level system entities, network propagation and individual causal discovery. (2) To evaluate the necessity and effectiveness of integrating the individual and topological causal discovery, we developed two variants: REASON-I, which only keeps the individual causal discovery, and REASON-T, which solely keeps the topological causal discovery. (3) To verify the efficacy of hierarchical GNN-based causal discovery, we replaced the causal discovery component of REASON with PC, C-LSTM, Dynotears, and GOLEM, respectively, to implement model variants denoted as REASON-P, REASON-C, REASON-D, and REASON-G. All experiments were conducted on a server running Ubuntu 18.04.5 with Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz, 4-way GeForce RTX 2080 Ti GPUs, and 192 GB memory. In addition, all methods were implemented using Python 3.8.12 and PyTorch 1.7.1.

4.2 Performance Evaluation

4.2.1 Overall Performance. Table 1, Table 2, and Table 3 present the overall performance of all models, where a larger value indicates better performance. We have two key observations: First, REASON can significantly outperform all the baselines on all three datasets. For example, compared to the second-best method, REASON can improve PR@10, MAP@10, and MRR by at least 21.9%, 15.7%, and 6.29%, respectively. The underlying driver is that REASON can capture more complex malfunctioning effects by integrating individual and topological analyses, and learning interdependent causal networks. Second, GNN is the best baseline model that outperforms others on most datasets. A possible explanation is that graph neural networks can facilitate the learning of non-linear causal relations among system entities via message passing. Thus, the experimental results on three datasets demonstrate the superiority of REASON in locating root causes over other baselines.

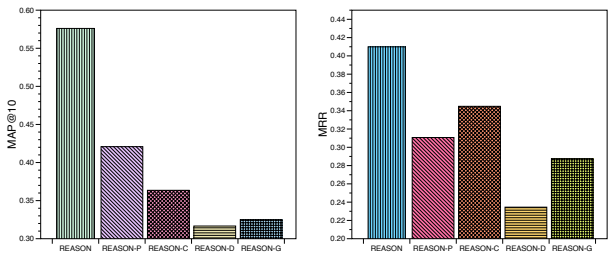


(a) Precision@K for SWaT (b) MAP@K for SWaT

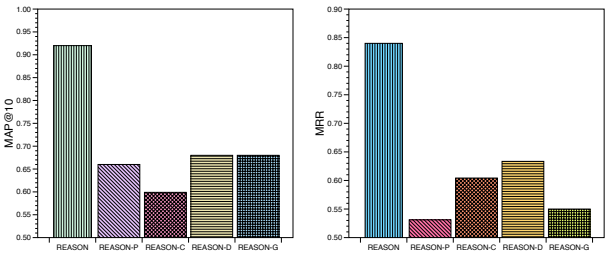


(c) Precision@K for AIOps (d) MAP@K for AIOps

Figure 4: Ablation studies of REASON.



(a) MAP@10 for SWaT (b) MRR for SWaT



(c) MAP@10 for AIOps (d) MRR for AIOps

Figure 5: The impact of hierarchical GNNs.

4.2.2 Influence of Network Propagation. Here, we applied our network propagation mechanism (see Section 3.1.2) to the causal structures learned by each baseline model to evaluate its effect on performance. The results are shown in Table 4 and Table 5. Our first finding is that network propagation can always improve the model performance for all models on all datasets. This observation strongly supports our assumption that network propagation is beneficial for capturing the propagation patterns of malfunctioning effects, resulting in a superior root cause localization performance. Moreover, we observe that across all models, network propagation yields greater performance enhancement on AIOps than the other two datasets. A possible reason is that AIOps contains explicit invoking relations among different pods, resulting in learning stronger causation compared with SWaT and WADI. Thus, this experiment indicates that network propagation is an important module for keeping an excellent performance for root cause localization.

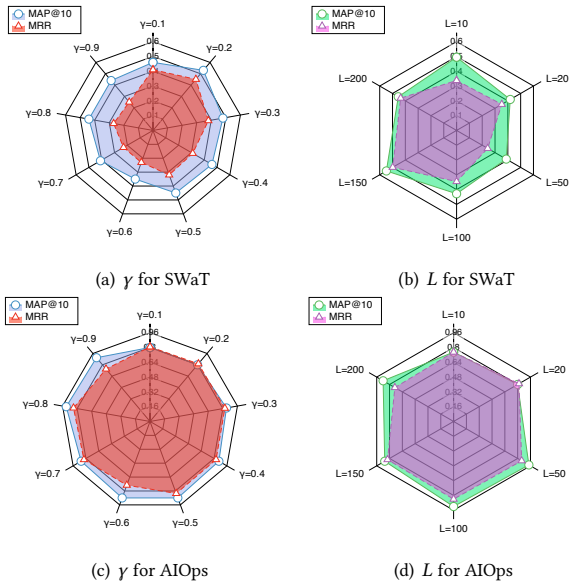
4.2.3 Ablation studies of REASON. Figure 4 shows ablation studies of REASON to examine the necessity of each technical component using PR@K and MAP@K. We can find that REASON significantly outperforms REASON-N on SWaT[3]. The underlying driver is that since REASON-N focuses on modeling causation among low-level entities only, using such causal structures, REASON-N is unable to capture cross-network propagation patterns of malfunctioning effects, leading to worse model performance. The second finding is that REASON is superior to both REASON-T and REASON-I in most cases. This observation indicates that integrating individual and topological causal discovery results can sufficiently capture the fluctuation and propagation patterns of malfunctioning effects for precisely locating root causes. Thus, each technical component of REASON is indispensable for keeping excellent model performance.

4.2.4 Impact of Hierarchical GNN-based Causal Discovery. Figure 5 evaluates the effectiveness of the proposed hierarchical GNN-based causal discovery method. Our key observations are two-fold. First, we find that REASON significantly outperforms all model variants. The underlying driver is that the message-passing mechanism of GNN can learn more robust non-linear causal relations through sharing neighborhood information. Moreover, REASON-P outperforms REASON-C across all datasets in terms of MAP@10, while the result is the opposite in terms of MRR. A possible explanation is that PC learns more causal relations between system entities than C-LSTM. As a result, REASON-P is able to identify more actual root causes by propagating on the learned causal structures, but their ranks are not at the top owing to more root cause candidates.

4.2.5 Parameter Analysis. We investigated the integration parameter γ and the number of layers L in GNN. γ controls the contribution of the individual and topological causal discovery for root cause localization. The number of layers L in GNN impacts the learning scenario of interdependent causal structures. Figure 6 presents our parameter analysis results. It can be seen that although the optimal γ values for different datasets vary, REASON can achieve optimal or near-optimal results on all three datasets using a similar small γ value. For instance, the best value of γ for SWaT and AIOps is 0.2 and 0.8, respectively. But when we use $\gamma = 0.1$, compared with the optimal results, the MRR value only decreased by 0.01 on SWaT and 0.04 on AIOps, respectively. This indicates that although the propagation of malfunctioning effects varies amongst datasets, the topological component contributes more to the model performance than the individual component, which further supports our findings in Section 4.2.3. Thus, in most cases, a small γ value (*e.g.*, $\gamma = 0.1$) would be a good choice. Second, when the number of GNN layers rose, we did not observe improved model performance. The reason is too many GNNs may cause the information of each node to become highly similar, hindering the learning of robust causation.

4.2.6 A Case Study. Finally, we conducted a case study to show the learned interdependent causation by using the system failure of AIOps on September 1, 2021. The detailed data collection procedure is as follows: First, the operators deployed a microservice system on three servers that are *compute-2*, *infra-1*, and *control-plane-1*. Then,

[3]Similar results can be observed on WADI data



(a) γ for SWaT (b) L for SWaT

(c) γ for AIOps (d) L for AIOps

Figure 6: Parameter analysis of REASON.

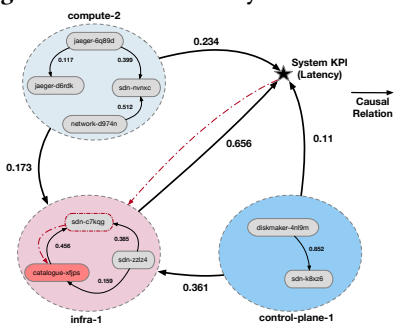


Figure 7: Interdependent causal graphs learned from AIOps dataset. Applications/pods are denoted by solid nodes, in which the red solid node is the root cause. The numbers on each edge with black color indicate the causal score. The red dashed line reflects the backtracing process of the root cause.

they sent requests periodically to the pod *sdn-c7kqg* to observe the system's latency. Next, to simulate the system attack, the operators used an *openssl* command to make the pod *catalogue-xfjp* have an extremely high CPU load, which affected some other pods on different servers, and eventually caused the system fault. Finally, the operators collected all entity metrics of all system entities.

Based on the collected metrics data, we applied REASON to learn the interdependent causation between system entities and the system KPI, which reflects the real operation circumstances. Figure 7 shows the learned interdependent causal graphs using the CPU Usage metric. According to it, *infra-1* server is the one most likely to increase in system latency. In this server, *catalogue-xfjp* is the root cause, whose negative effects propagate to '*sdn-c7kqg*, resulting in the malfunction of *infra-1*. This observation shows that REASON can precisely locate the root causes and provide an explanation for the located outcomes.

5 RELATED WORK

Root Cause Analysis (RCA), also known as fault localization, focuses on identifying the root causes of system failures/faults from symptom observations [43]. In recent years, many domain-specific RCA approaches [9, 11, 17, 19, 42, 47] have been proposed for maintaining the robustness of complex systems in various domains. Different from the existing works, the proposed REASON framework is a generic RCA approach that analyzes the surveillance multi-variate time series data from both individual and topological perspectives. Moreover, REASON captures the interdependent properties present in many complex systems to enhance RCA performance.

Causal Discovery in Time Series aims to learn causal relationships from observational time series data [5]. Existing methods can be broadly classified into four categories: (i) Granger causality approaches [31, 48], in which the causation is assessed based on whether one time series is helpful in predicting another; (ii) Constraint-based approaches [16, 39, 46], in which a causal structure is learned based on the conditional independence test and v-structure rules; (iii) Noise-based approaches [21, 37], in which the causation is depicted by equations that reflect the causation between different variables and noises; (iv) Score-based approaches [7, 36], in which a causal structure's quality is assessed by a scoring function. REASON belongs to the score-based causal discovery category. But different from existing works, our paper proposed a hierarchical GNN-based method to learn interdependent causation from multi-variable time series.

Interdependent Networks are often referred to as network of networks (NoN), in which complex networks interact and influence one another [20, 32, 49]. Numerous real-world systems exhibit such structural and dynamical features that differ from those observed in isolated networks. To overcome the limitation of prior efforts on isolated graph analysis, in recent years, increasing research efforts have been focused on interdependent networks and their applications [14, 34, 40]. For example, Ni *et al.* employed interdependent networks to illustrate the academic influence of scholars based on their research area and publications [34]. However, there are two key differences between REASON and other previous works: 1) Existing works only consider physical or statistical correlations, but not causation. 2) Existing interdependent networks are constructed using domain knowledge or system rules. REASON can auto-discover the interdependent causation from monitoring metrics data for RCA.

6 CONCLUSION

In this paper, we investigated the challenging problem of root cause localization in complex systems with interdependent network structures. We proposed REASON, a generic framework for root cause localization through mining interdependent causal relations and propagation patterns of faulted effects. Hierarchical graph neural networks were used to capture non-linear intra-level and inter-level causation and to improve causal discovery among system entities via message transmission. We conducted comprehensive experiments on three real-world datasets to evaluate the proposed framework. The experimental results validate the effectiveness of our work and the importance of capturing interdependent structures for root cause localization. An interesting direction for further exploration in the future would be incorporating other sources of data, such as system logs, with the time series data for root cause analysis in real-world complex systems.

REFERENCES

[1] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. 2017. WADI: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*. 25–28.

[2] M Hadi Amini, Kianoosh G Boroojeni, SS Iyengar, Panos M Pardalos, Frede Blaabjerg, and Asad M Madni. 2019. Sustainable interdependent networks II. *Studies in systems, decision and control* (2019), 167.

[3] M Hadi Amini, Ahmed Imteaj, and Panos M Pardalos. 2020. Interdependent networks: A data science perspective. *Patterns* 1, 1 (2020), 100003.

[4] Bjørn Andersen and Tom Fagerhaug. 2006. *Root cause analysis: simplified tools and techniques*. Quality Press.

[5] Charles K Assaad, Emilie Devijver, and Eric Gaussier. 2022. Survey and Evaluation of Causal Discovery Methods for Time Series. *Journal of Artificial Intelligence Research* 73 (2022), 767–819.

[6] Jan Beirlant, Yuri Goegebeur, Johan Segers, and Jozef L Teugels. 2004. *Statistics of extremes: theory and applications*. Vol. 558. John Wiley & Sons.

[7] Alexis Bellot, Kim Branson, and Mihaela van der Schaar. 2021. Neural graphical modelling in continuous-time: consistency guarantees and algorithms. In *International Conference on Learning Representations*.

[8] Stephen A Billings. 2013. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons.

[9] Álvaro Brandón, Marc Solé, Alberto Huélamo, David Solans, María S Pérez, and Victor Muntés-Mulero. 2020. Graph-based root cause analysis for service-oriented and microservice architectures. *Journal of Systems and Software* 159 (2020), 110432.

[10] 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.

[11] Alfonso Capozzoli, Fiorella Lauro, and Imran Khan. 2015. Fault detection analysis using data mining techniques for a cluster of smart office buildings. *Expert Systems with Applications* 42, 9 (2015), 4324–4338.

[12] Zhengzhang Chen, Kanchana Padmanabhan, Andrea M Rocha, Yekaterina Shpanskaya, James R Mihelcic, Kathleen Scott, and Nagiza F Samatova. 2012. SPICE: discovery of phenotype-determining component interplays. *BMC Systems Biology* 6, 1 (2012), 1–19.

[13] Wei Cheng, Kai Zhang, Haifeng Chen, Guofei Jiang, Zhengzhang Chen, and Wei Wang. 2016. Ranking causal anomalies via temporal and dynamical analysis on vanishing correlations. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 805–814.

[14] Arun Das, Joydeep Banerjee, and Arunabha Sen. 2014. Root Cause Analysis of Failures in Interdependent Power-Communication Networks. In *2014 IEEE Military Communications Conference*. 910–915.

[15] Boxiang Dong, Zhengzhang Chen, Hui Wang, Lu-An Tang, Kai Zhang, Ying Lin, Zhichun Li, and Haifeng Chen. 2017. Efficient discovery of abnormal event sequences in enterprise security systems. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 707–715.

[16] Doris Entner and Patrik O Hoyer. 2010. On causal discovery from time series data using FCI. *Probabilistic graphical models* (2010), 121–128.

[17] George K Fourlas and George C Karras. 2021. A survey on fault diagnosis methods for UAVs. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 394–403.

[18] Jianxi Gao, Daqing Li, and Shlomo Havlin. 2014. From a single network to a network of networks. *National Science Review* 1, 3 (2014), 346–356.

[19] Jiaping Gui, Ding Li, Zhengzhang Chen, Junghwan Rhee, Xusheng Xiao, Mu Zhang, Kangkook Jee, Zhichun Li, and Haifeng Chen. 2020. APTrace: A responsive system for agile enterprise level causality analysis. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1701–1712.

[20] Ait Mimoune Hamiche, Amine Boudghene Stambouli, and Samir Flazi. 2016. A review of the water-energy nexus. *Renewable and Sustainable Energy Reviews* 65 (2016), 319–331. https://doi.org/10.1016/j.rser.2016.07.020

[21] Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O Hoyer. 2010. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research* 11, 5 (2010).

[22] Emre Kıcıman and Lakshminarayanan Subramanian. 2005. Root cause localization in large scale systems. In *Proc. 1st Workshop on Hot Topics in Systems Dependability*.

[23] Maya Kosoff. 2022. One Amazon Employee's "Human Error" May Have Cost The Economy Millions. [EB/OL]. https://www.vanityfair.com/news/2017/03/one-amazon-employees-human-error-may-have-cost-the-economy-millions.

[24] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[25] Zeyan Li, Junjie Chen, Rui Jiao, Nengwen Zhao, Zhijun Wang, Shuwei Zhang, Yanjun Wu, Long Jiang, Leiqin Yan, Zikai Wang, Zhekang Chen, Wenchi Zhang, Xiaohui Nie, Kaixin Sui, and Dan Pei. 2021. Practical Root Cause Localization for Microservice Systems via Trace Analysis. In *2021 IEEE/ACM 29th International Symposium on Quality of Service*. 1–10. https://doi.org/10.1109/IWQOS52092.2021.9521340

[26] JinJin Lin, Pengfei Chen, and Zibin Zheng. 2018. Microscope: Pinpoint performance issues with causal graphs in micro-service environments. In *International Conference on Service-Oriented Computing*. Springer, 3–20.

[27] Dewei Liu, Chuan He, Xin Peng, Fan Lin, Chenxi Zhang, Shengfang Gong, Ziang Li, Jiayu Ou, and Zheshun Wu. 2021. MicroHECL: high-efficient root cause localization in large-scale microservice systems. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice*. IEEE, 338–347.

[28] Xueming Liu, H Eugene Stanley, and Jianxi Gao. 2016. Breakdown of interdependent directed networks. *Proceedings of the National Academy of Sciences* 113, 5 (2016), 1138–1143.

[29] Aditya P Mathur and Nils Ole Tippenhauer. 2016. SWaT: A water treatment testbed for research and training on ICS security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, 31–36.

[30] Yuan Meng, Shenglin Zhang, Yongqian Sun, Ruru Zhang, Zhilong Hu, Yiyin Zhang, Chenyang Jia, Zhaogang Wang, and Dan Pei. 2020. Localizing failure root causes in a microservice through causality inference. In *2020 IEEE/ACM 28th International Symposium on Quality of Service*. IEEE, 1–10.

[31] Meike Nauta, Doina Bucur, and Christin Seifert. 2019. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction* 1, 1 (2019), 312–340.

[32] M. Nekovee, Y. Moreno, G. Bianconi, and M. Marsili. 2007. Theory of rumour spreading in complex social networks. *Physica A: Statistical Mechanics and its Applications* 374, 1 (2007), 457–470. https://doi.org/10.1016/j.physa.2006.07.017

[33] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. 2020. On the role of sparsity and dag constraints for learning linear dags. *Advances in Neural Information Processing Systems* 33 (2020), 17943–17954.

[34] Jingchao Ni, Hanghang Tong, Wei Fan, and Xiang Zhang. 2014. Inside the atoms: ranking on a network of networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1356–1365.

[35] Jingchao Ni, Hanghang Tong, Wei Fan, and Xiang Zhang. 2015. Flexible and Robust Multi-Network Clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 835–844.

[36] Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. 2020. Dynotears: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1595–1605.

[37] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2013. Causal inference on time series using restricted structural equation models. *Advances in Neural Information Processing Systems* 26 (2013).

[38] James Pickands III. 1975. Statistical inference using extreme order statistics. *the Annals of Statistics* (1975), 119–131.

[39] Jakob Runge. 2020. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In *Conference on Uncertainty in Artificial Intelligence*. PMLR, 1388–1397.

[40] Davood Shiri and Vahid Akbari. 2021. Online Failure Diagnosis in Interdependent Networks. *Operations Research Forum* 2, 1 (2021), 10.

[41] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. 2017. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1067–1075.

[42] Jacopo Soldani and Antonio Brogi. 2022. Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–39.

[43] Marc Solé, Victor Muntés-Mulero, Annie Ibrahim Rana, and Giovani Estrada. 2017. Survey on models and techniques for root-cause analysis. *arXiv preprint arXiv:1701.08546* (2017).

[44] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. 2000. *Causation, prediction, and search*. MIT press.

[45] James H Stock and Mark W Watson. 2001. Vector autoregressions. *Journal of Economic perspectives* 15, 4 (2001), 101–115.

[46] Jie Sun, Dane Taylor, and Erik M Bollt. 2015. Causal network inference by optimal causation entropy. *SIAM Journal on Applied Dynamical Systems* 14, 1 (2015), 73–106.

[47] LuAn Tang, Hengtong Zhang, Zhengzhang Chen, Bo Zong, LI Zhichun, Guofei Jiang, and Kenji Yoshihira. 2019. Graph-based attack chain discovery in enterprise security systems. US Patent 10,289,841.

[48] A Tank, I Covert, N Foti, A Shojaie, and EB Fox. 2021. Neural Granger Causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[49] Dongjie Wang, Zhengzhang Chen, Jingchao Ni, Liang Tong, Zheng Wang, Yanjie Fu, and Haifeng Chen. 2023. Hierarchical Graph Neural Networks for Causal Discovery and Root Cause Localization. *arXiv preprint arXiv:2302.01987* (2023).

[50] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. 2018. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems* 31 (2018).