Online Multi-modal Root Cause Identification in Microservice Systems

Lecheng Zheng[1,2], Zhengzhang Chen[2,*], Haifeng Chen[2]
[1]University of Illinois Urbana-Champagin, [2]NEC Labs America
lecheng4@illinois.edu, zchen@nec-labs.com, haifeng@nec-labs.com

***Abstract*—Root Cause Analysis (RCA) is essential for pinpointing the root causes of failures in microservice systems. Traditional data-driven RCA methods are typically limited to offline applications due to high computational demands, and existing online RCA methods handle only single-modal data, overlooking complex interactions in multi-modal systems. In this paper, we introduce OCEAN, a novel online multi-modal causal structure learning method for root cause localization. OCEAN introduces a long-term temporal causal learning module with two encoders: one captures stable causal dependencies from historical data, while the other models short-term variations in the current batch data. We further design a multi-factor attention mechanism to analyze and reassess the relationships among different metrics and log indicators/attributes for enhanced online causal graph learning. Additionally, a contrastive mutual information maximization-based graph fusion module is developed to effectively model the relationships across various modalities. Extensive experiments on three real-world datasets demonstrate the effectiveness and efficiency of our proposed method.**

***Index Terms*—Root Cause Analysis, Microservice System, Multi-modal Learning.**

I. Introduction

Root Cause Analysis (RCA) is crucial for identifying the underlying causes of system failures and ensuring the high performance of microservice systems [1, 2]. Traditional manual root cause analysis is labor-intensive, costly, and error-prone, given the complexity of microservice systems and the extensive volume of data involved. Consequently, effective and efficient root cause analysis methods are vital for pinpointing failures in complex microservice systems and mitigating potential financial losses when system faults occur. Prior works leveraging causal discovery have focused on constructing causal or dependency graphs [3, 4, 2], which reveal causal links among system entities and key performance indicators to trace underlying faults.

Despite significant advances, most of these approaches are designed for offline use and face challenges with real-time implementation in microservice systems due to high computational demands. To address this, Wang *et al.* [1] introduced an online RCA method that decouples state-invariant and state-dependent information and incrementally updates the causal graph. Li *et al.* [5] developed a causal Bayesian network that leverages system architecture knowledge to mitigate potential biases toward new data. However, these online RCA methods are limited to handling single-modal data. Recently, multi-modal data, such as system metrics and logs, are commonly collected from microservice systems, revealing the complex nature of system failures [6]. For instance, failures such as "Database Query Failures" might be overlooked if only system metrics are considered, whereas issues like "Disk Space Full" are more effectively identified through combined analysis of metrics and logs. This underscores the importance of using multi-modal data for a thorough understanding of system failures. By integrating information from various sources, we can detect the abnormal patterns of system failures that is evident when analyzing single-modal data.

To bridge this gap, this paper proposes an online multi-modal causal structure learning method for RCA in microservice systems. Given the system KPI and multi-modal data (metrics and log data), our objective is to construct an online multi-modal causal graph that identifies the top k system entities most relevant to the system KPI. Three major challenges arise in this task: (C1) **Enabling Long-term Temporal Causal Relationship Learning in the Online Setting**: Existing temporal/sequential modeling techniques [7, 8, 9, 1, 6], (e.g., RNNs, Transformers, and LLMs) are computationally expensive and limited to short-term dependencies. However, some system faults, such as Distributed Denial of Service (DDoS) attacks, may persist for extended periods. Effectively capturing these long-term temporal dependencies in the online setting is crucial for efficiently identifying various types of system faults. (C2) **Capturing the Correlation of Multi-dimensional Factors**: Existing RCA approaches [3, 10, 6] often analyze abnormal patterns from multiple factors individually, such as CPU usage or memory usage from system metrics and frequency or golden signal from system logs, overlooking potential relationships among these factors from both modalities. Furthermore, these methods often consider all factors as equally important; however, in real applications, certain factors prove to be considerably more crucial than others. It is vital, therefore, to reassess the contributions of each factor to the learning of causal structures. (C3) **Learning Multi-modal Causal Structures**: Effectively capturing the relationships between different modalities in an online setting is crucial. Simply combining causal graphs from individual modalities can be problematic, especially if one modality is of lower quality.

To tackle these challenges, we introduce OCEAN, Online Multi-modal Causal Structure LEArNing, for root cause identification in microservice systems. Specifically, we design a long-term temporal causal relationship learning module with two encoders, one for capturing the long-term temporal

. *Corresponding author

dependencies and invariant causal relation for historical data and another for modeling the small changes within current batch data. We further develop a multi-factor attention mechanism to analyze the correlations among various factors and reassess their importance for causal graph learning. Additionally, we propose a contrastive mutual information estimation technique to model the relationships of different modalities. Our contributions can be summarized as follows:

- We introduce a novel online framework for multi-modality root cause analysis.
- We propose long-term temporal causal relationship learning module with two encoders, aiming to efficiently capture long-term temporal dependencies and causal relations for both historical data and current batch data.
- We develop graph fusion techniques with contrastive multi-modal learning to model the relationships between different modalities and assess their importance.
- Extensive experiments on three real-world datasets demonstrate the effectiveness and efficiency of our proposed method.

II. Preliminary and Related Work

Key Performance Indicator (KPI). In a microservice system, KPIs serve as invaluable metrics for assessing the effectiveness and productivity of the architecture [11]. They play an indispensable role in monitoring and managing different aspects of microservices to uphold optimal performance levels. Common KPIs encompass latency and service response time. High values in these metrics typically indicate suboptimal performance or potential failure.

Entity Metrics. Entity metrics are the measurable time-series attributes that provide insights into the performance and status of services within a system [12]. These entities encompass various components such as physical machines, containers, virtual machines, and pods. In microservice architectures, typical entity metrics include CPU utilization, memory usage, disk I/O activity, packet transmission rate, and etc. These metrics are extensively employed to detect anomalous behavior and pinpoint potential causes of system failures in microservice environments [6, 13].

Root Cause Analysis. Current root cause analysis (RCA) methods can be categorized into two main branches: single-modal RCA methods and multi-modal RCA methods. Single-modal RCA methods primarily investigate causal relationships among system components using one type of data only [13, 2, 10, 1], while multi-modal RCA methods [14, 15, 6] benefit from leveraging the rich data sources to achieve better performance. Recently, large language model (LLM)-based approaches have emerged as a new research direction for learning causal relations in root cause identification, owing to the success of LLMs in tackling complex tasks [7, 8, 16, 9, 17, 18]. Unlike existing RCA methods, this paper addresses the online multi-modal RCA problem by uniquely modeling long-term temporal dependencies while simultaneously capturing the cross-modal correlation of multiple factors.

TABLE I
Notation Table

$\boldsymbol{X}_M^0$	the historical system metric (offline data)
$\boldsymbol{X}_M^i$	the i-th batch of the system metric (streaming data)
$\boldsymbol{X}_L^0$	the historical system log (offline data)
$\boldsymbol{X}_L^i$	the i-th batch of system log (streaming data)
T_1	the length of the historical metric data
T_2	the length of the batch for the system metric
$n-1$	the number of system entities
$\mathcal{T}$	the total number of batches
d_M	the number of different system metric features
d_L	the number of different system log features
$\mathbf{y}$	the system Key Performance Indicator
$\mathcal{G} = \{\mathcal{V}, \mathcal{A}\}$	the causal graph
$\mathcal{A}$	the adjacency matrix in the causal graph

III. Methodology

In this section, we first present the problem statement and then introduce OCEAN, an online causal structural learning method designed to identify root causes using multi-modal data. We propose three modules to tackle the challenges outlined in the introduction: long-term temporal causal relationship learning, contrastive multi-modal learning and network propagation-based root cause identification module. The overview of OCEAN is provided in Figure 1.

A. Problem Statement

Let $\mathcal{X}_M = \{\mathbf{X}_M^0, \mathbf{X}_M^1, \ldots, \mathbf{X}_M^{\mathcal{T}}\}$ denote the multivariate time-series data of **system metrics** across $n-1$ entities, where $\mathbf{X}_M^0 \in \mathbb{R}^{(n-1)\times d_M \times T_1}$ represents a large historical dataset and $\mathbf{X}_M^i \in \mathbb{R}^{(n-1)\times d_M \times T_2}$ for $i \in [1, \mathcal{T}]$ are smaller sequential streaming batches. Here, d_M is the number of metric features, and T_1 and T_2 denote the temporal lengths of the historical and streaming segments, respectively. Similarly, $\mathcal{X}_L = \{\mathbf{X}_L^0, \mathbf{X}_L^1, \ldots, \mathbf{X}_L^{\mathcal{T}}\}$ denotes the multivariate time-series data of **system logs**, where each $\mathbf{X}_L^i \in \mathbb{R}^{(n-1)\times d_L \times T}$ is obtained by preprocessing log records into structured time-series format, and d_L is the number of log features. The **system KPI** is represented as $\mathbf{y} = \{\mathbf{y}^0, \mathbf{y}^1, \ldots, \mathbf{y}^{\mathcal{T}}\}$, where $\mathbf{y}^0 \in \mathbb{R}^{T_1}$ and $\mathbf{y}^i \in \mathbb{R}^{T_2}$ correspond to historical and batch KPI data, respectively. We consider n nodes in total: $n-1$ microservice entities and one KPI node. The causal structure among them is modeled by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ denotes the set of vertices and $\mathcal{A} \in \mathbb{R}^{n\times n}$ is the adjacency matrix, with $\mathcal{A}_{ij} \neq 0$ indicating a causal influence from node i to node j. For unified processing, we replicate the KPI d_M times and concatenate it with the metric and log data, forming $\hat{\mathbf{X}}_M^0 \in \mathbb{R}^{n\times d_M \times T_1}$, $\hat{\mathbf{X}}_M^i \in \mathbb{R}^{n\times d_M \times T_2}$, and $\hat{\mathbf{X}}_L^0 \in \mathbb{R}^{n\times d_L \times T_1}$, $\hat{\mathbf{X}}_L^i \in \mathbb{R}^{n\times d_L \times T_2}$. In the **offline setting**, only the historical data ($\mathbf{X}^0$, $\mathbf{y}^0$) are available for model training, while in the **online setting**, streaming batches ($\mathbf{X}^i$, $\mathbf{y}^i$) are also used for root cause analysis. Finally, we employ the **Multivariate Singular Spectrum Analysis (MSSA)** model [19], a state-of-the-art method for online anomaly detection, to trigger the root cause analysis process. We summarize the notations in Table I.

B. Long-term Temporal Causal Relationship Learning

To effectively capture temporal causal relationships among system entities and KPIs, we adopt the Vector Autoregression

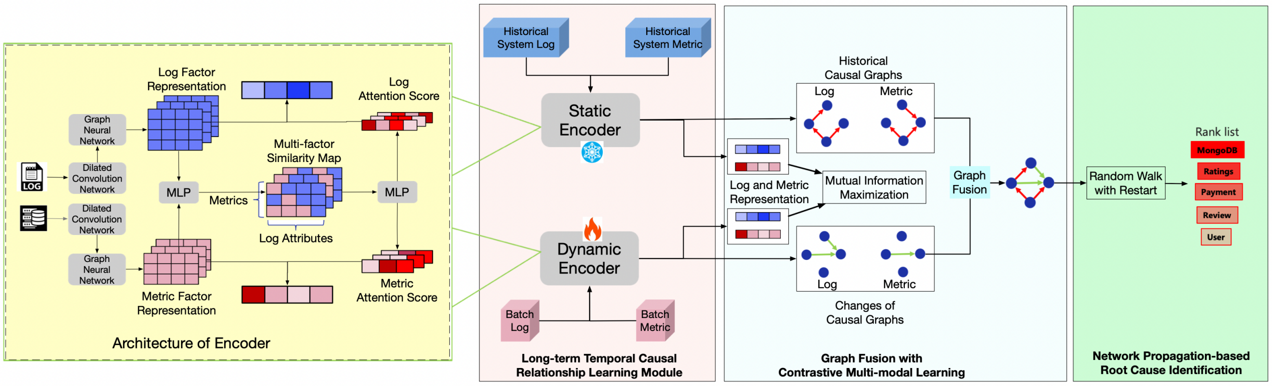

Fig. 1. The overview of the proposed framework OCEAN with three main modules: long-term temporal causal relationship learning, graph fusion with contrastive multi-modal learning, and network propagation-based root cause localization. Both static encoder and dynamic encoder are trained in the offline stage but only dynamic encoder will be finetuned with batch (streaming) data during the online stage.

(VAR) model [20], a classical yet powerful framework for modeling dynamic interactions in multivariate time series. Given the two-dimensional matrix $\mathbf{X}_{M,i} \in \mathbb{R}^{n\times T_1}$ representing the i^{th} system metric, the model predicts future values as:

$$\begin{aligned}
\hat{\mathbf{X}}^t_{M,i} &= \mathbf{W}^1\mathbf{X}^{t-1}_{M,i} + \mathbf{W}^2\mathbf{X}^{t-2}_{M,i} + \cdots + \mathbf{W}^{t-1}\mathbf{X}^{1}_{M,i} + \boldsymbol{\epsilon},\\
\mathcal{L}_{\text{var}} &= \sum_{i=1}^{d_M} \|\mathbf{X}^t_{M,i} - F(\hat{\mathbf{X}}^t_{M,i}, \mathcal{A}, \theta)\|^2, \qquad (1)
\end{aligned}$$

where $\hat{\mathbf{X}}^t_{M,i}$ denotes the predicted values for the i^{th} metric, $\mathbf{W}^k \in \mathbb{R}^{n\times n}$ are lag-specific weight matrices, $\boldsymbol{\epsilon}$ is the residual error, and $F(\cdot)$ represents a graph neural network (GNN) [21] parameterized by θ. The learnable adjacency matrix $\mathcal{A}$ encodes causal relationships among entities and the KPI. Although a t^{th}-order VAR model captures long-range dependencies, it becomes computationally expensive as t increases [22]. Similarly, recurrent architectures (e.g., RNNs) and Transformers [23] often suffer from high computational overhead, limiting their suitability for online root cause analysis. To address these challenges, we propose a dual-encoder framework consisting of a *Static Encoder* and a *Dynamic Encoder* to efficiently learn long-term temporal dependencies and causal structures for *historical* and *streaming* data, respectively. The Static Encoder extracts stable, invariant causal relationships among entities in the offline stage, while the Dynamic Encoder incrementally adapts the learned causal structure during online updates.

a) Static Encoder: In microservice systems, each entity is associated with multiple metric and log indicators (e.g., CPU usage, memory usage, log frequency, golden signals, etc.). Conventional RCA methods typically analyze each factor independently, overlooking cross-factor dependencies and their varying importance under different abnormal patterns. To address this limitation, we design the Static Encoder to jointly model temporal dependencies and multi-factor correlations across two modalities (metrics and logs) using attention-based multi-factor learning [24, 23]. Given historical metric data $\hat{\mathbf{X}}^0_M$ and log data $\hat{\mathbf{X}}^0_L$, we first capture temporal dependencies via a gated temporal convolutional network (TCN) [25]:

$$g(\mathbf{x}, \mathbf{f}) = \mathbf{x} * \mathbf{f} = \sum_{\tau=0}^{K-1} \mathbf{f}(\tau)\cdot\mathbf{x}(t - d\times\tau), \qquad (2)$$

$$\mathbf{H}^0_v = \tanh(g(\hat{\mathbf{X}}^0_v, \mathbf{f}_1)) \odot \sigma(g(\hat{\mathbf{X}}^0_v, \mathbf{f}_2)), \qquad (3)$$

where $\mathbf{f} \in \mathbb{R}^K$ represents the 1-D kernel, $\mathbf{f}_1, \mathbf{f}_2 \in \mathbb{R}^K$ are 1-D convolution kernels, d is the dilation factor, $\odot$ denotes the Hadamard product, $\sigma(\cdot)$ and $\tanh(\cdot)$ are the sigmoid and tanh functions respectively. The output $\mathbf{H}^0_v \in \mathbb{R}^{n\times d_v\times T_3}$ represents temporally encoded features for modality $v \in \{M, L\}$. Stacking dilated convolution layers exponentially increases the receptive field, enabling efficient long-term modeling with reduced computational cost. We validate the efficiency of dilated convolutional operations in our experiments (see Subsection IV-B0a) by comparing their computational costs with those of VAR-based methods.

To explore the correlation of different factors from two modalities and then assess the contribution of each factor to causal structure learning, we compute a multi-factor similarity matrix for the j^{th} system entity:

$$\mathbf{C}^0_j = \tanh(\mathbf{H}^0_M[j]\mathbf{W}^3(\mathbf{H}^0_L[j])^\top), \qquad (4)$$

where $\mathbf{W}^3 \in \mathbb{R}^{T_3\times T_3}$ is a learnable projection. The similarity matrix $\mathbf{C}^0_j \in \mathbb{R}^{d_L\times d_M}$ captures inter-factor relationships between metric and log modalities for the historical data. We then compute attention-based importance weights to quantify each factor's contribution to causal structure learning:

$$\begin{aligned}
\mathbf{Z}^0_L[j] &= \tanh(\mathbf{H}^0_L[j]\mathbf{W}^4 + \mathbf{H}^0_M[j]\mathbf{W}^5\mathbf{C}^0_j),\\
\mathbf{Z}^0_M[j] &= \tanh(\mathbf{H}^0_M[j]\mathbf{W}^5 + \mathbf{H}^0_L[j]\mathbf{W}^4(\mathbf{C}^0_j)^\top), \qquad (5)\\
\mathbf{a}^0_L[j] &= \text{softmax}(\mathbf{w}^6\mathbf{Z}^0_L[j]), \quad \mathbf{a}^0_M[j] = \text{softmax}(\mathbf{w}^7\mathbf{Z}^0_M[j]),
\end{aligned}$$

where $\mathbf{W}^4, \mathbf{W}^5 \in \mathbb{R}^{T_3\times T_3}$ and $\mathbf{w}^6, \mathbf{w}^7 \in \mathbb{R}^{T_4}$. The attention vectors $\mathbf{a}^0_v[j]$ measure the importance of each factor by encoding information from both modalities, capturing rich relationships for multi-modal and multi-dimensional data. With these attention vectors, the weighted modality representations are obtained as:

$$\hat{\mathbf{H}}_v^0[j] = \sum_{k=1}^{d_v} \mathbf{a}_v^0[j,k] \cdot \mathbf{H}_v^0[j,k], \tag{6}$$

and then passed through a multi-layer perceptron (MLP) for factor recovery:

$$\mathbf{O}_v^0 = \mathrm{MLP}^0(\hat{\mathbf{H}}_v^0). \tag{7}$$

Overall, the multi-factor learning process (Eqs. (4), (5), (6) and (7)) is summarized as:

$$\hat{\mathbf{O}}_v^0 = MFL(\hat{\mathbf{H}}_v^0). \tag{8}$$

To learn causal relationships among entities, we apply a message-passing GNN (GraphSAGE [26]) to mimic fault propagation through a message-passing mechanism:

$$\tilde{\mathbf{X}}_v^0 = \sigma_2(\mathcal{A}_{\text{old}}(\hat{\mathbf{O}}_v^0 \oplus \mathbf{N}_v^0)\mathbf{W}^1), \tag{9}$$
$$\mathbf{N}_v^0[j] = \frac{1}{|\mathcal{N}_j|} \sum_{k \in \mathcal{N}_j} \hat{\mathbf{O}}_v^0[k],$$

where $\mathbf{W}^1$ is the weight matrix, σ_2 denotes ReLU, $\oplus$ denotes concatenation, $\mathcal{A}_{\text{old}}$ is the historical causal graph, $\mathcal{N}_j$ represents node entity j's neighbors, and $\mathbf{N}_v^0$ aggregates neighbor information. $\tilde{\mathbf{X}}_v^0$ predicts future values based on previous lagged data $\hat{\mathbf{X}}_v^0$, leveraging temporal dependencies captured by dilated convolutional neural networks. Finally, the training objective minimizes the forecasting error:

$$\mathcal{L}_t = \frac{1}{n(d_L + d_M)} \sum_v \sum_{j=1}^{n} \sum_{k=1}^{d_v} \mathbf{a}_v^0[j,k] \|\hat{\mathbf{X}}_v^0[j,k] - \tilde{\mathbf{X}}_v^0[j,k]\|^2. \tag{10}$$

The $\mathbf{a}_v^0[j,k]$ emphasize influential factors during temporal forecasting, reinforcing interpretable causal learning.

b) Dynamic Encoder.: The *Dynamic Encoder* extends the same architecture to model streaming data in the online setting, enabling adaptive causal updates. It aims to efficiently model the long-term temporal dependencies and the causal relations among system entities and KPIs for the streaming data in the online setting. Given the current batch data $\hat{\mathbf{X}}_v^i$, temporal dependencies are encoded as:

$$\mathbf{H}_v^i = \tanh(g(\hat{\mathbf{X}}_v^i, \mathbf{f}_3)) \odot \sigma(g(\hat{\mathbf{X}}_v^i, \mathbf{f}_4)), \tag{11}$$
$$\hat{\mathbf{O}}_v^i = MFL(\hat{\mathbf{H}}_v^i), \tag{12}$$

where $\mathbf{f}_3, \mathbf{f}_4$ are dilated convolutional kernels. We then propagate messages through the updated causal graph:

$$\tilde{\mathbf{X}}_v^i = \sigma_2((\mathcal{A}_{\text{old}} + \Delta\mathcal{A}_v)(\hat{\mathbf{O}}_v^i \oplus \mathbf{N}_v^i)\mathbf{W}^2), \tag{13}$$
$$\mathbf{N}_v^i[j] = \frac{1}{|\mathcal{N}_j|} \sum_{k \in \mathcal{N}_j} \hat{\mathbf{O}}_v^i[k], \tag{14}$$

where $\Delta\mathcal{A}_v \in \mathbb{R}^{n \times n}$ is added to capture incremental causal changes in the current batch. Finally, we jointly minimize the reconstruction losses from historical and streaming data:

$$\mathcal{L}_t = \frac{1}{n(d_L + d_M)} \sum_v \sum_{j=1}^{n} \sum_{k=1}^{d_v} \mathbf{a}_v^i[j,k] \|\hat{\mathbf{X}}_v^i[j,k] - \tilde{\mathbf{X}}_v^i[j,k]\|^2$$

By integrating Eq. (15) with the message-passing mechanism in Eq. (13), the model continuously refines the causal adjacency matrix $\tilde{A} = \mathcal{A}_{\text{old}} + \Delta\mathcal{A}_v$, capturing dynamic causal relations such as $\mathbf{X} \rightarrow \mathbf{y}$, where $\mathbf{X}$ denotes a potential root cause and $\mathbf{y}$ is the system KPI.

C. Graph Fusion with Contrastive Multi-modal Learning

To address the challenges of multi-modal learning (as discussed in Challenge **C3** in Section I), we propose to enhance the relatedness between two modalities through **contrastive mutual information maximization**. Given the representations of historical data $\hat{\mathbf{H}}_v^0$ and current batch data $\hat{\mathbf{H}}_v^i$ extracted from both metric and log data, we maximize the mutual information between these modalities:

$$\mathcal{L}_{\text{MI}} = \mathcal{I}_\phi(\hat{\mathbf{H}}_M^0, \hat{\mathbf{H}}_L^0) + \mathcal{I}_\phi(\hat{\mathbf{H}}_M^i, \hat{\mathbf{H}}_L^i), \tag{15}$$

where $\mathcal{I}_\phi$ is the mutual information parameterized by a neural network ϕ. Following the InfoNCE-style contrastive loss [27], we approximate the mutual information by its lower bound:

$$\mathcal{I}_\phi(\hat{\mathbf{H}}_M^0, \hat{\mathbf{H}}_L^0) := \frac{1}{n} \sum_{j=1}^{n} \log \frac{\mathrm{sim}(\phi(\hat{\mathbf{H}}_M^0[j]), \phi(\hat{\mathbf{H}}_L^0[j]))}{\sum_k \mathrm{sim}(\phi(\hat{\mathbf{H}}_M^0[j]), \phi(\hat{\mathbf{H}}_L^0[k]))}, \tag{16}$$

where $\mathrm{sim}(a,b) = \exp\left(\frac{ab^\top}{\|a\|\|b\|}\right)$ represents the exponential of cosine similarity between two entity representations a and b.

To generate the causal graph for the current batch, directly summing the graphs from both modalities may yield overly dense or cyclic structures. This problem is amplified when one modality is of lower quality, as treating both modalities equally can obscure critical causal patterns. To mitigate this issue, we estimate the relative importance of the two modalities based on the correlation of multiple metrics. Using the similarity map for the current batch (*i.e.*, $\mathbf{C}_j^i$), we compute the modality importance and fuse the two causal graphs as follows:

$$\mathbf{C}_j^i = \tanh\big(\mathbf{H}_M^i[j]\mathbf{W}^8(\mathbf{H}_L^i[j])^\top\big),$$
$$s_M = \frac{\sum_j \sum_l \exp\big(\sum_k \mathbf{C}_j^i[l,k]\big)}{\sum_j \Big[\sum_l \exp\big(\sum_k \mathbf{C}_j^i[l,k]\big) + \sum_k \exp\big(\sum_l \mathbf{C}_j^i[l,k]\big)\Big]},$$
$$\mathcal{A} = (1 - s_M)\big(\mathcal{A}_{\text{old}} + \Delta\mathcal{A}_L\big) + s_M\big(\mathcal{A}_{\text{old}} + \Delta\mathcal{A}_M\big),$$

where $\mathbf{W}^8$ is a learnable weight matrix. This adaptive fusion mechanism ensures that higher-quality modalities contribute more to the causal graph update while maintaining structural sparsity and stability.

Optimization. The overall training objective is defined as:

$$\mathcal{L}_{\text{sparse}} = \|\Delta\mathcal{A}_L\|_1 + \|\Delta\mathcal{A}_M\|_1,$$
$$\mathcal{L} = -\mathcal{L}_{MI} + \lambda_1\mathcal{L}_t + \lambda_2\mathcal{L}_{\text{sparse}} + \lambda_3 h(\mathcal{A}), \tag{17}$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm that enforces sparsity on the changes of the adjacency matrices. The sparsity regularization $\mathcal{L}_{\text{sparse}}$ encourages only a limited number of edge updates across modalities. The acyclicity constraint $h(\mathcal{A}) = \text{tr}(e^{\mathcal{A}\odot\mathcal{A}}) - n = 0$ holds if and only if $\mathcal{A}$ is a directed acyclic graph [28], where $\odot$ denotes the Hadamard product. λ_1, λ_2, and λ_3 are positive hyperparameters balancing the contribution of each component.

Online Finetuning. During the offline training stage, both the static and dynamic encoders are jointly optimized using Eq. (17). In the online adaptation stage, the static encoder is frozen while the dynamic encoder is finetuned to efficiently adapt to streaming data with reduced computational overhead.

D. Network Propagation-based Root Cause Identification

Malfunction effects often propagate from the root cause to its neighboring entities, implying that the immediate neighbors of anomalous KPIs may not necessarily be the true root causes. To accurately identify the root cause, we first derive a transition probability matrix based on the learned causal graph $\mathcal{G}$ and then employ a random walk with restart (RWR) algorithm [29] to simulate the propagation of malfunction signals:

$$\mathbf{P}_{ij} = \frac{\mathcal{A}_{j,i}}{\sum_{k=1}^{n} \mathcal{A}_{k,i}}, \tag{18}$$

where $\mathbf{P}$ is the normalized adjacency matrix. During the random walk process, the model periodically restarts from the KPI node with probability $c \in [0,1]$ to explore alternative propagation paths. The RWR update equation is formulated as:

$$\mathbf{r}_{t+1} = (1-c)\mathbf{P}\mathbf{r}_t + c\mathbf{r}_0, \tag{19}$$

where $\mathbf{r}_t$ denotes the probability distribution over nodes at step t, and $\mathbf{r}_0$ is the initial distribution concentrated on the target KPI node. Upon convergence, the stationary distribution $\mathbf{r}_{t+1}$ reflects the influence score of each entity, and the top-k entities with the highest scores are identified as the most probable root causes.

Stopping Criterion. As the online RCA process progresses across multiple data batches, both the inferred causal structure and the corresponding root cause rankings tend to stabilize. To avoid redundant computation, we introduce an adaptive stopping mechanism based on the *rank-biased overlap* (RBO) metric [30], which quantifies the similarity between two ranked lists while emphasizing higher-ranked items. Given the root cause lists from consecutive batches, R_{t-1} and R_t, their similarity is computed as:

$$\gamma = \text{RBO}(R_{t-1}, R_t), \tag{20}$$

where $\gamma \in [0,1]$. A larger γ indicates a higher degree of stability in the root cause rankings. The online RCA process terminates automatically when the similarity score γ exceeds a predefined threshold.

IV. Experiments

In this section, we evaluate the effectiveness of our proposed OCEAN by comparing it with state-of-the-art root cause analysis techniques. Additionally, we conduct a case study and an ablation study to further validate the assumptions outlined in the previous sections.

A. Experimental Setup

Datasets. We evaluate the performance of OCEAN using three public real-world datasets: (1) **Product Review**[*] [31]: A microservice system dedicated to online product reviews, encompassing 234 pods deployed across 6 cloud servers. It recorded four system faults between May 2021 and December 2021. (2) **Online Boutique**[†] [14]: A microservice system designed for e-commerce, including five system faults. (3) **Train Ticket** [14]: A microservice system for railway ticketing services, also with five system faults. All three datasets contain two modalities: system metrics and system logs.

Evaluation Metrics. We evaluate the effectiveness of the proposed model with three widely-used metrics [10, 32]: (1) **Precision@K (PR@K)**: This metric measures the probability that the top-K predicted root causes are accurate. (2) **Mean Average Precision@K (MAP@K)**: It assesses the top-K predicted causes from an overall perspective. (3) **Mean Reciprocal Rank (MRR)**: This metric evaluates the ranking capability of the models. (4) **Time**: the training time (in seconds) for each batch of data.

Baselines. We compare OCEAN with eight methods: single modality RCA methods, including Dynotears [28],C-LSTM [33], REASON [10], multi-modality RCA method, i.e., MULAN [6], hypothesis based methods, PC [34], ϵ-Diagnosis [35], BARO [36], one online RCA method (i.e., CORAL [1]).

Reproducibility. All experiments are conducted on a desktop running Ubuntu 18.04.5 with an Intel(R) Xeon(R) Silver 4110 CPU @2.10GHz and one 11GB GTX2080 GPU. We use the Adam as the optimizer and we train the model for 100 iterations at each batch. We use two layers of dilated convolutional operations in the experiment. As for the stopping criteria, we terminate the identification process if the similarity γ between the current batch and the previous batch is greater than 0.9 for three consecutive times. Each experiment is run with one trial. For Product Review dataset, we set the size of historical metric and log data to 8-hour intervals and each batch is set to be a 10-minute interval. **Due to page limitation, we include the additional experiments, such as the experiment on Train Ticket dataset, the comparison with physical graph, time complexity analysis, three-modalities analysis, comparison with different architecture, and parameter analysis in a more complete version online**[‡].

B. Performance Evaluation

a) Experimental Results: In this subsection, we present the performance evaluation on Tables II and III for various methods. Considering that many baseline methods (*e.g.*, PC, C-LSTM, REASON, Dynotears, and Baro) are designed for the single-modality scenario, we assess their performance in both

[*]. https://lemma-rca.github.io/docs/data.html

[†]. https://github.com/IntelligentDDS/Nezha

[‡]. https://arxiv.org/abs/2410.10021

TABLE II
RESULTS ON PRODUCT REVIEW DATASET W.R.T DIFFERENT METRICS.

Modality	Model	PR@1	PR@5	PR@10	MRR	MAP@3	MAP@5	MAP@10	Time (s)
Metric Only	PC	0	0	0	0.034	0	0	0	225.19
	Dynotears	0	0.25	0.50	0.092	0	0.05	0.175	390.37
	C-LSTM	0.25	0.5	0.5	0.409	0.417	0.45	0.475	1482.01
	REASON	0.25	**1.00**	**1.00**	0.563	0.583	0.75	0.875	247.87
	CORAL	0.50	**1.00**	**1.00**	0.750	0.833	0.90	0.950	146.46
	Baro	0.50	0.50	0.75	0.537	0.500	0.50	0.550	283.48
	ϵ-Diagnosis	0	0	0.25	0.038	0	0	0.050	367.13
Log Only	PC	0	0	0	0.043	0	0	0	93.98
	Dynotears	0	0	0.25	0.058	0	0	0.075	142.26
	C-LSTM	0	0	0.25	0.059	0	0	0.075	602.92
	REASON	0	0	0.5	0.088	0	0	0.100	129.17
	CORAL	0	0	0.5	0.118	0	0	0.200	50.29
	Baro	0.25	0.25	0.25	0.286	0.250	0.25	0.250	138.94
	ϵ-Diagnosis	0	0	0.25	0.039	0	0	0.050	149.23
Multi-Modality	PC	0	0	0.25	0.054	0	0	0.075	300.26
	Dynotears	0	0.25	0.5	0.114	0	0.05	0.225	426.78
	C-LSTM	0.25	0.5	0.5	0.341	0.250	0.35	0.425	1808.76
	REASON	0.5	**1.00**	**1.00**	0.687	0.667	0.80	0.900	303.5
	MULAN	0.75	**1.00**	**1.00**	0.833	0.833	0.90	0.950	255.74
	CORAL	0.75	**1.00**	**1.00**	0.875	0.917	0.95	0.975	186.73
	Baro	0.50	0.75	**1.00**	0.587	0.500	0.60	0.700	307.26
	ϵ-Diagnosis	0	0	0.25	0.042	0	0	0.075	402.33
	OCEAN	**1.00**	**1.00**	**1.00**	**1.000**	**1.000**	**1.00**	**1.000**	**20.16**

TABLE III
RESULTS ON ONLINE BOUTIQUE W.R.T DIFFERENT METRICS.

Modality	Model	PR@1	PR@3	PR@5	MRR	MAP@2	MAP@3	MAP@5	Time (s)
Metric Only	PC	0.2	0.4	0.6	0.390	0.3	0.333	0.40	5.25
	Dynotears	0.2	0.4	0.4	0.344	0.2	0.267	0.32	14.56
	C-LSTM	0	0.4	0.8	0.300	0.1	0.200	0.44	20.75
	REASON	0.4	0.8	**1.0**	0.617	0.5	0.600	0.76	3.23
	CORAL	0.2	**1.0**	**1.0**	0.600	0.6	0.733	0.84	2.99
	Baro	0	0.8	**1.0**	0.417	0	0.467	0.68	3.46
	ϵ-Diagnosis	0	0.6	**1.0**	0.323	0	0.267	0.52	5.21
Log Only	PC	0	0.4	0.6	0.257	0.1	0.200	0.32	3.88
	Dynotears	0	0.2	0.6	0.207	0	0.067	0.24	10.23
	C-LSTM	0	0.4	0.6	0.267	0.1	0.200	0.36	15.07
	REASON	0.2	0.8	0.8	0.458	0.3	0.467	0.60	2.39
	CORAL	0.2	0.6	**1.0**	0.457	0.3	0.400	0.60	2.04
	Baro	0	0.6	0.8	0.308	0	0.267	0.48	2.57
	ϵ-Diagnosis	0	0.4	0.4	0.208	0	0.133	0.24	3.47
Multi-Modality	PC	0.4	0.8	**1.0**	0.573	0.4	0.533	0.68	6.78
	Dynotears	0.2	0.6	**1.0**	0.467	0.3	0.400	0.64	16.38
	C-LSTM	0.2	0.4	**1.0**	0.450	0.3	0.333	0.60	22.66
	REASON	0.4	**1.0**	**1.0**	0.667	0.6	0.733	0.84	4.51
	MULAN	0.4	0.8	**1.0**	0.617	0.5	0.600	0.76	4.96
	CORAL	0.4	**1.0**	**1.0**	0.700	0.7	0.800	0.88	3.63
	Baro	0.2	**1.0**	**1.0**	0.567	0.2	0.667	0.80	4.88
	ϵ-Diagnosis	0	0.8	**1.0**	0.383	0	0.400	0.64	6.51
	OCEAN	**0.6**	**1.0**	**1.0**	**0.800**	**0.8**	**0.867**	**0.92**	**1.84**

single-modality scenarios (*e.g.*, system metrics only or system logs only) and the multi-modality case. We consider system logs as additional system metrics to enable the performance measurement of these single-modality RCA methods in the multi-modality scenario. We calculate an average ranking score based on the evaluation of different system metrics as the final result for all single-modality methods and OCEAN. Our observations are as follows: (1) Compared to single-modality scenarios, most baseline methods benefit from leveraging multi-modality data across three distinct datasets. (2) As an online RCA method, CORAL outperforms all of the offline RCA methods with respect to seven metrics. (3). OCEAN consistently outperforms all baseline methods across the three datasets. (4). The online RCA methods CORAL and OCEAN have less training time compared with the offline RCA methods, while OCEAN further reduces its computational cost to 1/9 compared with CORAL. The reduced computational cost is attributed to the efficiency of dilated convolutional operation and the design of the multi-factor attention mechanism. Notice that CORAL first learns the causal graph for each factor individually and then fuses the causal graphs, which is computationally expensive in the online setting. Notably, OCEAN exhibits a remarkable improvement in MRR on the Product Review dataset, excelling the second competitor (*i.e.*, CORAL) by 12.5%. Moreover, OCEAN outperforms CORAL by 20% and 10% with respect to PR@1 and MAP@3, respectively. This is attributed to the assessment of multiple factors and the exploration of the correlation among different modalities.

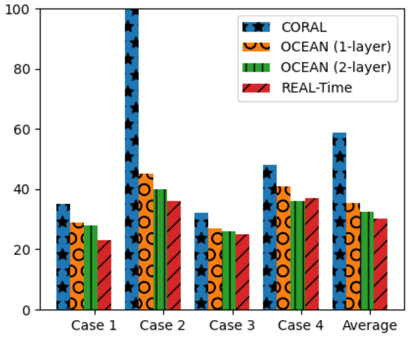

Fig. 2. Identification time for four cases as well as the average identification time.

TABLE IV
Ablation study on three datasets w.r.t MRR.

Model	Product Review	Online Boutique	Train Ticket
OCEAN	**1.00**	**0.8**	**0.381**
OCEAN-F	0.75	**0.8**	0.331
OCEAN-M	0.875	0.7	0.320
OCEAN-S	0.833	0.7	0.345

b) Case Study: In this subsection, we evaluate the promptness of two online RCA methods on the Product Review dataset, CORAL and OCEAN, as shown in Figure 2. Note that we also evaluate the effectiveness of the long-term temporal causal structure learning module by varying the number of dilated convolutional layers in OCEAN, specifically comparing configurations with one and two layers. In Figure 2, the y-axis represents the batch index at which an RCA method meets the stopping criteria, and the real-time marker indicates the actual system failure time. A lower batch index value signifies faster identification of the ground-truth root cause by the RCA method. Notably, CORAL did not successfully rank the ground-truth root cause first in case 2, so we use the total number of batches to represent its detection time for a fair comparison. Our observations reveal that CORAL experiences about a 10-epoch delay relative to real-time in most cases, whereas OCEAN (2-layer) achieves quicker detection than OCEAN (1-layer). This improvement confirms our hypothesis that adding more dilated convolutional layers enhances the model's ability to capture longer temporal dependencies, as discussed in Subsection III-B.

c) Ablation Study: In this subsection, we evaluate the effectiveness of individual components within the objective function of OCEAN (Eq. 17). Specifically, we define OCEAN-F and OCEAN-M as variants that lack the multi-factor attention learning module and the contrastive multi-modal learning module, respectively, while OCEAN-S removes the sparse constraint. The results, shown in Table IV, indicate a significant performance degradation when any component is omitted. Specifically, removing the multi-factor attention module results in 25% and 5% performance drop on the Product Review dataset and Train Ticket dataset, respectively. Eliminating the contrastive multi-modal learning module leads to 12.5% reduction on the Product Review dataset. These findings underscore the importance of each component in maintaining OCEAN's high performance.

V. Conclusion

In this paper, we investigate the challenging problem of online multi-modal root cause localization in microservice systems. We introduce OCEAN, a novel online causal structure learning framework designed to effectively identify root causes using diverse data sources. OCEAN utilizes a dilated convolutional neural network to capture long-term temporal dependencies and employs graph neural networks to establish causal relationships among system entities and key performance indicators. Additionally, we develop a multi-factor attention mechanism to evaluate and refine the contributions of various factors to the causal graph. Furthermore, OCEAN incorporates a contrastive mutual information maximization-based graph fusion module to enhance interactions between different modalities and optimize their mutual information. The effectiveness of OCEAN is validated through extensive experiments on three real-world datasets, demonstrating its robustness and efficiency.

References

[1] D. Wang, Z. Chen, Y. Fu, Y. Liu, and H. Chen, "Incremental causal graph learning for online root cause analysis," in *SIGKDD*, 2023, pp. 2269–2278.

[2] Z. Li, J. Chen, R. Jiao, N. Zhao, Z. Wang, S. Zhang, Y. Wu, L. Jiang, L. Yan, Z. Wang *et al.*, "Practical root cause localization for microservice systems via trace analysis," in *29th IWQOS*. IEEE, 2021, pp. 1–10.

[3] A. Ikram, S. Chakraborty, S. Mitra, S. K. Saini, S. Bagchi, and M. Kocaoglu, "Root cause analysis of failures in microservices through causal discovery," in *NeurIPS*, 2022.

[4] S. Lu, B. Rao, X. Wei, B. Tak, L. Wang, and L. Wang, "Log-based abnormal task detection and root cause analysis for spark," in *ICWS*. IEEE, 2017, pp. 389–396.

[5] M. Li, Z. Li, K. Yin, X. Nie, W. Zhang, K. Sui, and D. Pei, "Causal inference-based root cause analysis for online service systems with intervention recognition," in *SIGKDD*. ACM, 2022, pp. 3230–3240.

[6] L. Zheng, Z. Chen, J. He, and H. Chen, "MULAN: multi-modal causal structure learning and root cause analysis for microservice systems," in *WWW 2024*. ACM, 2024, pp. 4107–4116.

[7] Y. Chen, H. Xie, M. Ma, Y. Kang, X. Gao, L. Shi, Y. Cao, X. Gao, H. Fan, M. Wen *et al.*, "Automatic root cause analysis via large language models for cloud incidents," in *EuroSys*, 2024, pp. 674–688.

[8] S. Shan, Y. Huo, Y. Su, Y. Li, D. Li, and Z. Zheng, "Face it yourselves: An llm-based two-stage strategy to localize

configuration errors via logs," in *SIGSOFT*, 2024, pp. 13–25.

[9] B. Zhou, X. Li, T. Liu, K. Xu, W. Liu, and J. Bao, "Causalkgpt: industrial structure causal knowledge-enhanced large language model for cause analysis of quality problems in aerospace product manufacturing," *Advanced Engineering Informatics*, vol. 59, p. 102333, 2024.

[10] D. Wang, Z. Chen, J. Ni, L. Tong, Z. Wang, Y. Fu, and H. Chen, "Interdependent causal networks for root cause localization," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*. ACM, 2023, pp. 5051–5060.

[11] D. Podgórski, "Measuring operational performance of osh management system–a demonstration of ahp-based selection of leading key performance indicators," *Safety science*, vol. 73, pp. 146–166, 2015.

[12] J. Bogner, S. Wagner, and A. Zimmermann, "Automatically measuring the maintainability of service-and microservice-based systems: a literature review," in *IWSM Mensura*, 2017, pp. 107–115.

[13] J. Soldani and A. Brogi, "Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–39, 2022.

[14] G. Yu, P. Chen, Y. Li, H. Chen, X. Li, and Z. Zheng, "Nezha: Interpretable fine-grained root causes analysis for microservices on multi-modal observability data," in *the 31st FSE*, 2023, pp. 553–565.

[15] C. Hou, T. Jia, Y. Wu, Y. Li, and J. Han, "Diagnosing performance issues in microservices with heterogeneous data source," in *ISPA*. IEEE, 2021, pp. 493–500.

[16] D. Goel, F. Husain, A. Singh, S. Ghosh, A. Parayil, C. Bansal, X. Zhang, and S. Rajmohan, "X-lifecycle learning for cloud incident management using llms," in *32nd ACM FSE*, 2024, pp. 417–428.

[17] D. Roy, X. Zhang, R. Bhave, C. Bansal, P. Las-Casas, R. Fonseca, and S. Rajmohan, "Exploring llm-based agents for root cause analysis," in *32nd ACM FSE*, 2024, pp. 208–219.

[18] Z. Wang, Z. Liu, Y. Zhang, A. Zhong, L. Fan, L. Wu, and Q. Wen, "Rcagent: Cloud root cause analysis by autonomous agents with tool-augmented large language models," in *CIKM*, 2024.

[19] A. Alanqary, A. Alomar, and D. Shah, "Change point detection via multivariate singular spectrum analysis," *NeurIPS*, vol. 34, pp. 23 218–23 230, 2021.

[20] J. H. Stock and M. W. Watson, "Vector autoregressions," *Journal of Economic perspectives*, vol. 15, no. 4, pp. 101–115, 2001.

[21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR 2017*. OpenReview.net, 2017.

[22] C.-C. Lin, A. Jaech, X. Li, M. R. Gormley, and J. Eisner, "Limitations of autoregressive models and their alternatives," *arXiv preprint arXiv:2010.11939*, 2020.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS 30*, 2017, pp. 5998–6008.

[24] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *NeurIPS 29*, 2016, pp. 289–297.

[25] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *IJCAI 2019*. ijcai.org, 2019, pp. 1907–1913.

[26] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *NeurIPS*, vol. 30, 2017.

[27] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[28] R. Pamfil, N. Sriwattanaworachai, S. Desai, P. Pilgerstorfer, K. Georgatzis, P. Beaumont, and B. Aragam, "DYNOTEARS: structure learning from time-series data," in *The 23rd AISTATS*, vol. 108. PMLR, 2020, pp. 1595–1605.

[29] H. Tong, C. Faloutsos, and J. Pan, "Fast random walk with restart and its applications," in *Proceedings of ICDM 2006*. IEEE Computer Society, 2006, pp. 613–622.

[30] W. Webber, A. Moffat, and J. Zobel, "A similarity measure for indefinite rankings," *ACM Trans. Inf. Syst.*, vol. 28, no. 4, pp. 20:1–20:38, 2010.

[31] L. Zheng, Z. Chen, D. Wang, C. Deng, R. Matsuoka, and H. Chen, "Lemma-rca: A large multi-modal multi-domain dataset for root cause analysis," *arXiv preprint arXiv:2406.05375*, 2024.

[32] Y. Meng, S. Zhang, Y. Sun, R. Zhang, Z. Hu, Y. Zhang, C. Jia, Z. Wang, and D. Pei, "Localizing failure root causes in a microservice through causality inference," in *28th IWQoS*. IEEE, 2020, pp. 1–10.

[33] A. Tank, I. Covert, N. J. Foti, A. Shojaie, and E. B. Fox, "Neural granger causality," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4267–4279, 2022.

[34] T. Burr, "Causation, prediction, and search," *Technometrics*, vol. 45, no. 3, pp. 272–273, 2003.

[35] H. Shan, Y. Chen, H. Liu, Y. Zhang, X. Xiao, X. He, M. Li, and W. Ding, "ϵ-diagnosis: Unsupervised and real-time diagnosis of small-window long-tail latency in large-scale microservice platforms," in *WWW*, 2019, pp. 3215–3222.

[36] L. Pham, H. Ha, and H. Zhang, "Baro: Robust root cause analysis for microservices via multivariate bayesian online change point detection," *Proceedings of the ACM on Software Engineering*, vol. 1, no. FSE, pp. 2214–2237, 2024.