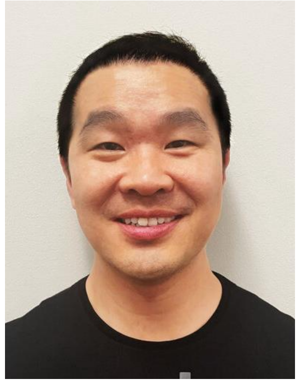
THE WEB CONFERENCE ACM

NEC Laboratories America

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

MULAN: Multi-modal Causal Structure Learning and Root Cause Analysis for Microservice Systems

Lecheng Zheng
(UIUC)
lecheng4@illinois.edu

Zhengzhang Chen
(NEC)
zchen@nec-labs.com

Jingrui He
(UIUC)
jingrui@illinois.edu

Haifeng Chen
(NEC)
haifeng@nec-labs.com

NSF

Artificial Intelligence
iSAIL Lab

Roadmap

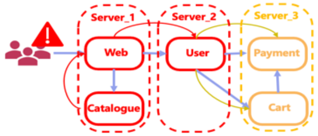
Background

Motivation Overview

I. Background

- ➢ Motivation
- ➢ Challenges
- ➢ Problem Definition

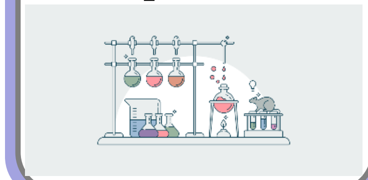
Methodology



II. Proposed Method

- ➢ Log-tailored Language Model
- ➢ Contrastive Multi-modal Causal Structure Learning
- ➢ Causal Graph Fusion with KPI-Aware Attention

Experiment



III. Experiments

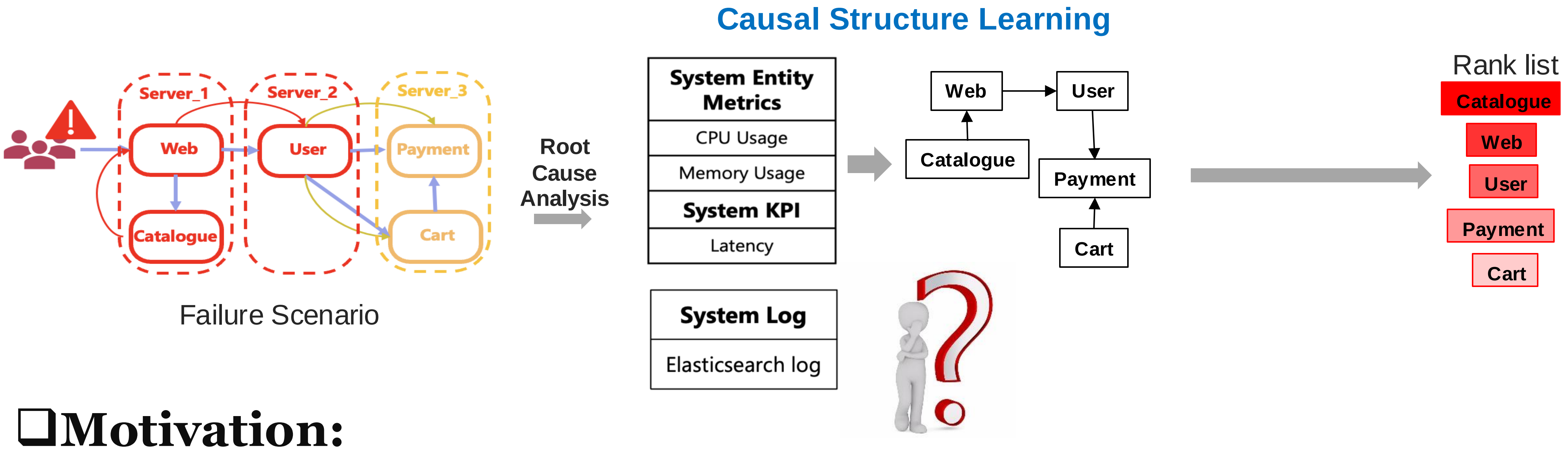
- ➢ Effectiveness

Conclusion

IV. Conclusion

Root Cause Analysis (RCA) in AIOps

❑Microservice example



❑Motivation:

- Most existing methods [1,2] only focus on system metrics and fail to handle multi-modal data.
- Only utilizing single modality may lead to **incomplete insights** and **overlook correlation** among different modalities [3,4].

- [1] Dongjie Wang, et al. Interdependent Causal Networks for Root Cause Localization. In SIGKDD 2023.
- [2] Azam Ikram, et al. Root Cause Analysis of Failures in Microservices through Causal Discovery. In NeurIPS 2022.
- [3] Chuanjia Hou, et al. Diagnosing Performance Issues in Microservices with Heterogeneous Data Source. In ISPA/BDCloud/SocialCom/SustainCom, 2021.
- [4] Guangba Yu, et al. Nezha: Interpretable Fine-Grained Root Causes Analysis for Microservices on Multi-modal Observability Data. In ESEC/FSE 2023.

Challenges

- ❑ C1: Learning effective representation of system logs for causal graph learning
  - Unstructured system logs lack formal grammar rules and extensively employ special tokens.
- ❑ C2: Learning causal structure from multi-modal data
  - Solely relying on data from a single modality fails to capture various abnormal patterns.
- ❑ C3: Assessing modality reliability
  - Low-quality data can obscure crucial patterns, making it a challenging task to identify root cause.

- W0523 07:54:43.595523 1 cacher.go:148] Terminating all watchers from cacher *build.BuildConfig
- I0523 07:54:43.622746 1 cacher.go:402] cacher (*user.User): initialized

C1: Unstructured system logs

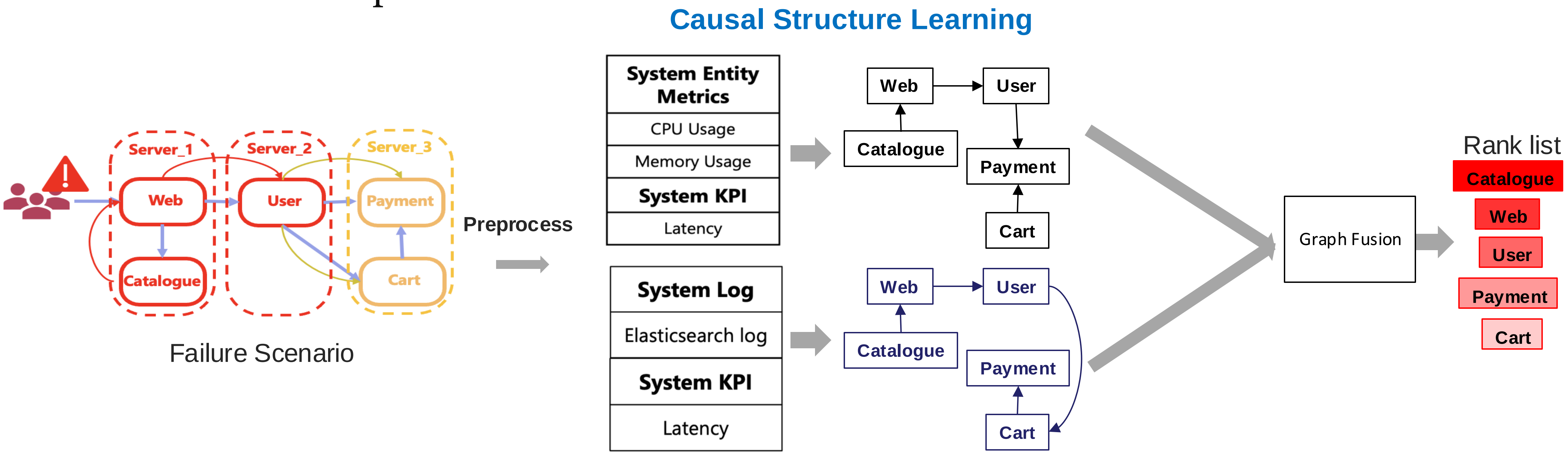
System Fault Type	System Metric	System Log
Database Query Failure	-	Error/Warning
Login Failure	-	Error/Warning
DDoS Attack	High CPU Utilization	-
Disk Space Full	High Disk Utilization	Error/Warning

C2: some abnormal patterns may exist in one modality.

C3: low-quality data can obscure crucial patterns.

Problem Definition

❑ Microservice example



❑ **Input:** System entity metrics X^M, system logs (*e.g.*, Elasticsearch logs) X^L, and system KPI (*i.e.*, multi-variate time series) y

❑ **Output:** Top-k possible root causes related to system failures and causal graph $\mathcal{G} = \{V, A\}$ for further system diagnosis

Roadmap

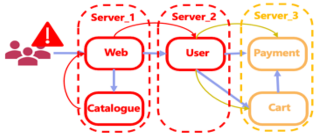
Background

Motivation Overview

I. Background

- ➢ Motivation
- ➢ Challenges
- ➢ Problem Definition

Methodology



II. Proposed Method

- ➢ Log-tailored Language Model
- ➢ Contrastive Multi-modal Causal Structure Learning
- ➢ Causal Graph Fusion with KPI-Aware Attention

Experiment

III. Experiments

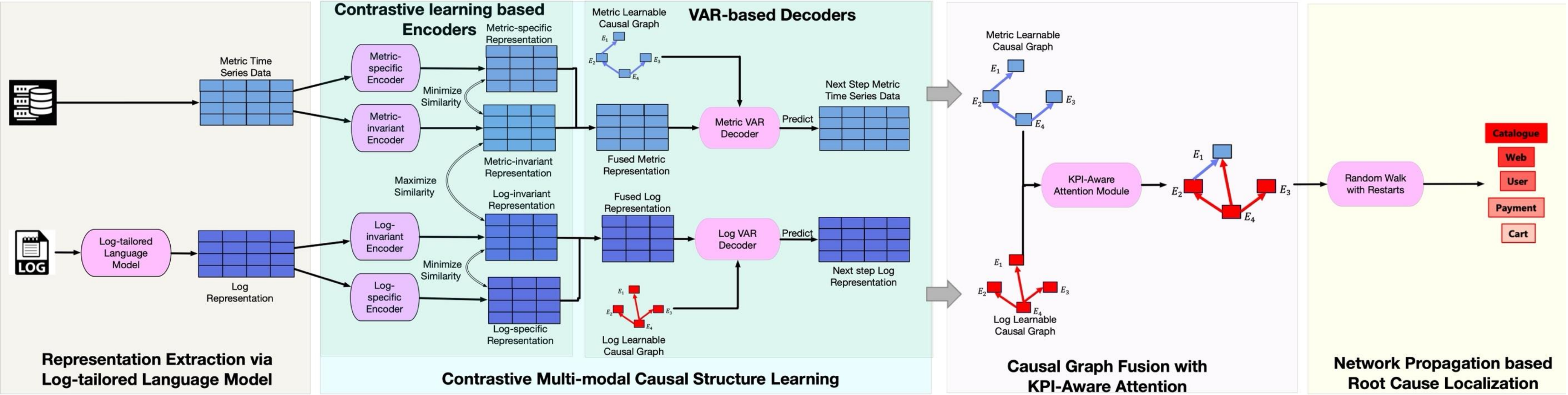
- ➢ Effectiveness

Conclusion

IV. Conclusion

Framework Overview

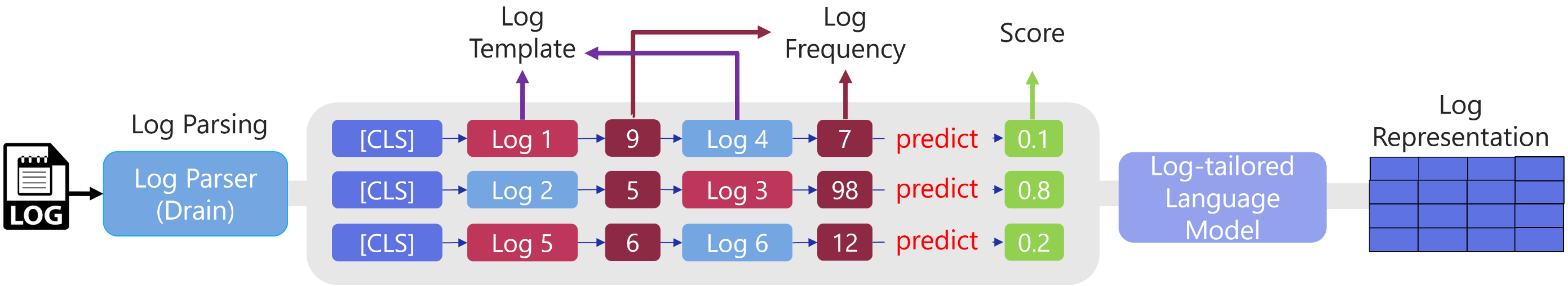
❑Multi-modal Causal Structure Learning (MULAN)



- A log-tailored language model to transform raw system logs into log time series data (addressing C1)
- A contrastive multi-modal causal structure learning module to extract both the modality-invariant and modality-specific representations and learn two causal graphs (addressing C2)
- A KPI-aware causal graph fusion module to assess the reliability of each modality and fuse the two causal graphs (addressing C3)

Log-tailored Language Model

- C1: Learning effective representation of system logs for causal graph learning
  – Unstructured system logs lack formal grammar rules and extensively employ special tokens.
- Solution:
  – We treat each log template as a token, and the log templates within a sequence are organized based on their first appearance timestamp in ascending order.
  – We consider the frequency of each unique log template, assuming that more frequently occurring log event templates carry more important information.



Contrastive Multi-modal Causal Structure Learning

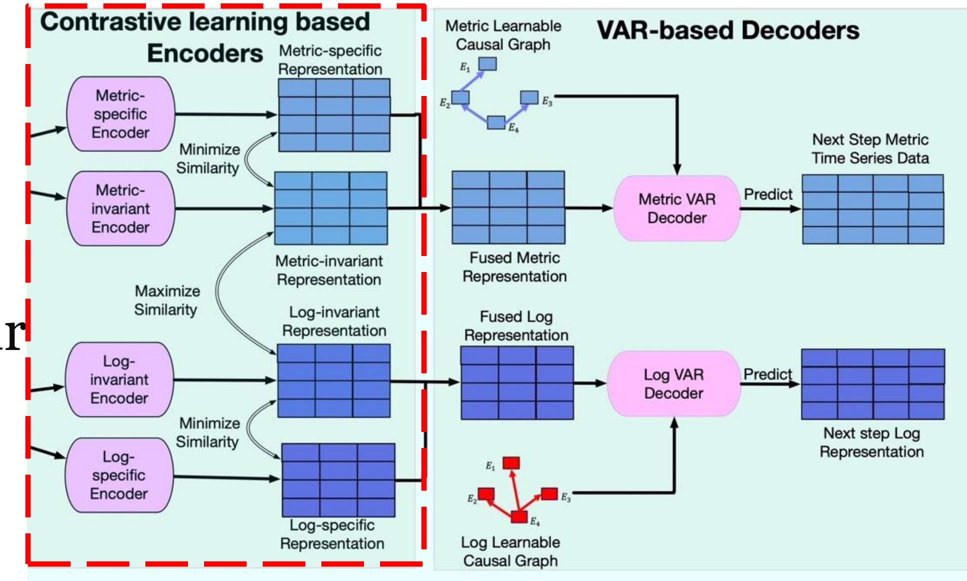
❑Contrastive Learning-based Encoders addressing C2:

- Modality-invariant representation: $R_c^v = E_c^v(X^v, A^v), v \in \{M, L\}$
- Modality-specific representation: $R_s^v = E_s^v(X^v, A^v), v \in \{M, L\}$
- The backbone of encoders $E_c^v(\cdot)$ and $E_s^v(\cdot)$ are GraphSage [7].
- A^v is the causal graph.

❑Mutual Information Maximization:

$$L_{node} = -\frac{1}{n}\sum_i \frac{sim(h_i^M, h_i^L)}{\sum_k sim(h_i^M, h_k^L)}$$

- $sim(a, b)$ is the exponential of cosine similarity measur
- $H^v = MLP^v(R_c^v)$ is the entity representation
- **Intuition:** Ensure mutual agreement between two modalities.



Contrastive Multi-modal Causal Structure Learning

- [7] William L. Hamilton, et al. Inductive Represen- tation Learning on Large Graphs. In NeruIPS 2017.

Contrastive Multi-modal Causal Structure Learning

❑Orthogonal Constraint:

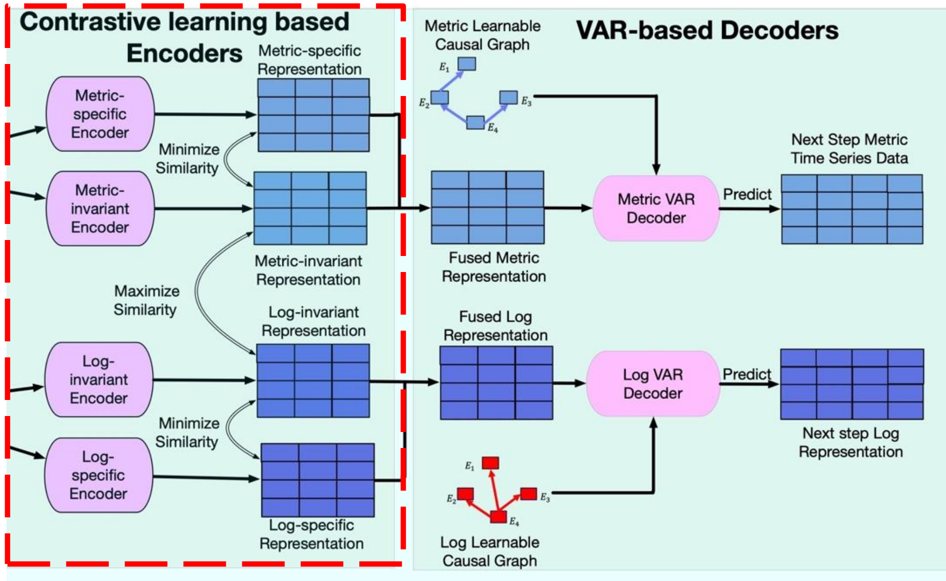
$$L_{orth} = \sum_{v\in\{M,L\}} \sum_{i=1}^{n} \left\| \left(R_{s,i}^{v}\right)^{T} R_{c,i}^{v} \right\|^{2}$$

– **Intuition:** Ensure no overlapping between modality-invariant representation and modality-specific representation.

❑Edge Prediction Loss:

– $L_{edge} = \sum_{v\in\{M,L\}} \sum_{i,j} \left\| G\left(e_{ij}^{v}\right) - A_{ij}^{v} \right\|^{2}$

– **Intuition:** The entity representation should contain enough information to predict the adjacency matrix of the causal graph.



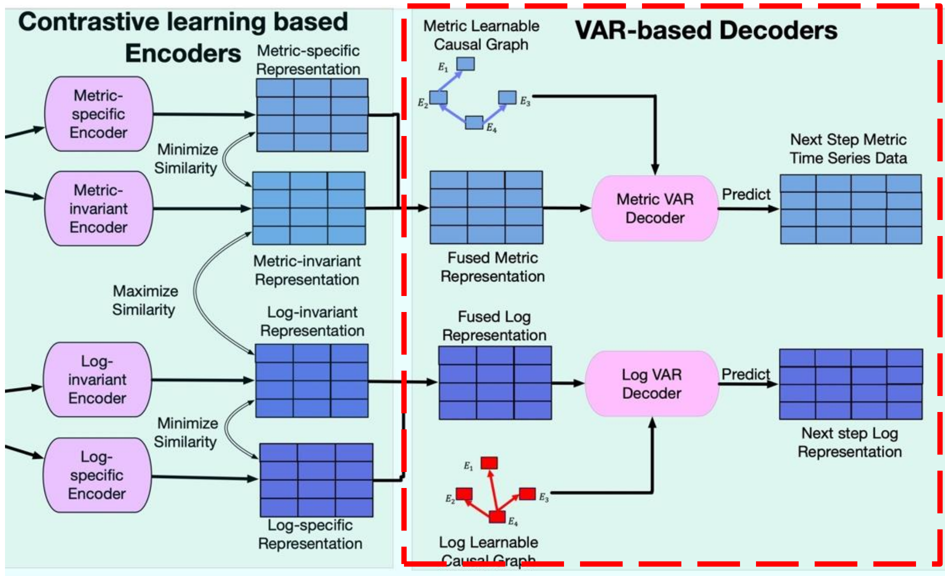
Contrastive Multi-modal Causal Structure Learning

❑VAR-based Decoders:

– We aim to predict the future value $\tilde{X}^v$ with the previous p-th lagged data $\hat{X}^v$ via VAR model:

$$L_{var} = \sum_{v \in \{M,L\}} \left\| \tilde{X}^v - D^v(R_c + R_s^v) \right\|^2$$

– **Intuition:** We aim to learn the causal relation among different entities via VAR model.



Contrastive Multi-modal Causal Structure Learning

KPI-Aware Causal Graph Fusion

- C3: How to alleviate the potential negative impact if the quality of one modality is not good enough?
- **Solution:** We propose to evaluate modality quality based on the correlation between node entity and KPI :

$$\alpha^v = \mathrm{softmax}_{v\in\{M,L\}}(\sum_i S_i^v)$$

– We measure the **cross correlation** between the node feature X^v and the KPI y:

$$S^v = \max_{p\in[0,\tau]} (X^v \odot y)(\tau) = \max_{p\in[0,\tau]} \int_t X^v(t+p)^T y(t)dt$$

– Where p is the time lag and τ is the max time lag.

– **Intuition:** For each modality $v \in \{M, L\}$, S^v measures the similarity between node and KPI with p time-lag, which provides the inference of the causality $X \rightarrow y$.

Root Cause Localization

- We use network propagation to mimic the propagation patterns of system malfunctions.
- Procedure:
  - We first derive the transition probability matrix based on the causal graph.

$$P_{ij} = \frac{(1-\beta)A_{j,i}}{\sum_{k=1}^{n} A_{j,i}}$$

    - $\beta \in [0,1]$ represents the probability of transitioning from one node to another.
  - We employ a random walk with restart method [8] to mimic the propagation patterns of malfunctions.

$$P_{t+1} = (1-c)P_t + cP_0$$

    - P_t denotes the jumping probability at the t-th step, P_0 is the initial starting probability, and $c \in [0,1]$ is the restart probability.

- [8] Hanghang Tong, et al. Fast Random Walk with Restart and Its Applications. In ICDM 2006.

Roadmap

Background

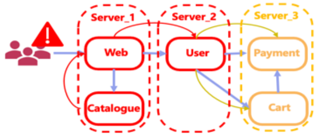

Motivation


Overview

I. Background

- ➢ Motivation
- ➢ Challenges
- ➢ Problem Definition

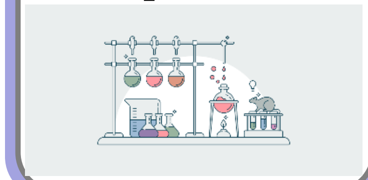
Methodology



II. Proposed Method

- ➢ Log-tailored Language Model
- ➢ Contrastive Multi-modal Causal Structure Learning
- ➢ Causal Graph Fusion with KPI-Aware Attention

Experiment



III. Experiments

- ➢ Effectiveness

Conclusion

IV. Conclusion

Experimental Setup

❑Datasets:

- Product Review: a microservice system, dedicated to online product reviews.
- Online Boutique: a microservice system designed for e-commerce
- Train Ticket: a microservice system for railway ticketing service

❑Baseline methods:

- PC [9]: a classic constraint-based causal discovery algorithm
- Dynotears [10]: a vector autoregression model constructing dynamic Bayesian network
- C-LSTM [11]: a LSTM based model capturing nonlinear Granger causality
- GOLEM [12]: a variant of NOTEARS relaxing the hard Directed Acyclic Graph constraint
- REASON [13]: An interdependent network model learning multi-level causal relationships
- Nezha [14]: A multi-modal method identifying root causes by detecting abnormal patterns

- [9] Tom Burr. 2003. Causation, Prediction, and Search. Technometrics 2003.
- [10] Roxana Pamfil, et al. DYNOTEARS: Structure Learning from Time-Series Data. In AISTATS 2020.
- [11] Alex Tank, et al. Neural Granger Causality. In TPAMI 2022.
- [12] Ignavier Ng, et al. On the Role of Sparsity and DAG Constraints for Learning Linear DAGs. In NeurIPS 2020.
- [13] Dongjie Wang, et al. Interdependent Causal Networks for Root Cause Localization. In SIGKDD 2023.
- [14] Guangba Yu, et al. Nezha: Interpretable Fine-Grained Root Causes Analysis for Microservices on Multi-modal Observability Data. In ESEC/FSE 2023.

Experimental Results

- (1) Most baseline methods demonstrate improved performance when leveraging multi-modality data across various metrics.
- (2) MULAN consistently outperforms all baseline methods across the three datasets.

Modality	Model	PR@1	PR@5	PR@10	MRR	MAP@3	MAP@5	MAP@10
Metric Only	Dynotears	0	0	0.50	0.070	0	0	0.075
	PC	0	0	0.25	0.053	0	0	0.050
	C-LSTM	0.25	0.75	0.75	0.474	0.5	0.25	0.675
	GOLEM	0	0	0.25	0.043	0	0	0.025
	REASON	0.75	**1.0**	**1.0**	0.875	0.917	0.95	0.975
Log Only	Dynotears	0	0	0.25	0.058	0	0	0.075
	PC	0	0	0.25	0.069	0	0	0.075
	C-LSTM	0	0	0.25	0.059	0	0	0.075
	GOLEM	0	0	0.25	0.058	0	0	0.075
	REASON	0	0.50	0.75	0.216	0.167	0.25	0.400
Multi-Modality	Dynotears	0	0	0.50	0.095	0	0	0.150
	PC	0	0	0.25	0.064	0	0	0.125
	C-LSTM	0.50	0.75	0.75	0.592	0.583	0.65	0.700
	GOLEM	0	0	0.25	0.065	0	0	0.050
	REASON	0.75	**1.0**	**1.0**	0.875	0.917	0.95	0.975
	Nezha	0	0.5	0.75	0.193	0.083	0.25	0.475
	MULAN	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Product Review Dataset

Modality	Model	PR@1	PR@3	PR@5	MRR	MAP@2	MAP@3	MAP@5
Metric Only	Dynotears	0.20	0.40	0.40	0.344	0.20	0.267	0.320
	PC	0.20	0.40	0.80	0.390	0.30	0.333	0.400
	C-LSTM	0	0.40	0.80	0.30	0.10	0.200	0.440
	GOLEM	0	0.40	0.80	0.291	0.20	0.267	0.360
	REASON	0.40	0.80	1.0	0.617	0.50	0.200	0.440
Log Only	Dynotears	0	0.20	0.60	0.207	0	0.067	0.240
	PC	0	0.40	0.60	0.257	0.10	0.200	0.320
	C-LSTM	0	0.40	0.60	0.267	0.10	0.200	0.360
	GOLEM	0	0.40	0.80	0.248	0	0.133	0.360
	REASON	0.20	0.80	0.80	0.458	0.30	0.467	0.600
Multi-Modality	Dynotears	0.20	0.60	1.0	0.467	0.30	0.400	0.640
	PC	0.40	0.80	1.0	0.573	0.40	0.533	0.680
	C-LSTM	0.20	0.40	1.0	0.450	0.30	0.333	0.600
	GOLEM	0.20	0.60	1.0	0.467	0.30	0.400	0.640
	REASON	0.40	1.0	1.0	0.667	0.60	0.733	0.840
	Nezha	0.60	1.0	1.0	0.767	0.70	0.800	0.880
	MULAN	**0.80**	**1.0**	**1.0**	**0.900**	**0.90**	**0.933**	**0.960**

Online Boutique Dataset

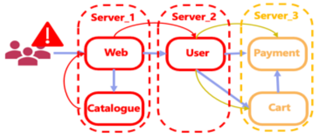
Roadmap

Background

I. Background

- ➢ Motivation
- ➢ Challenges
- ➢ Problem Definition

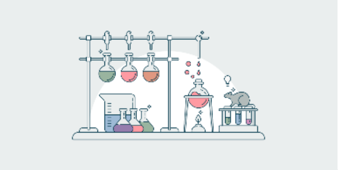
Methodology



II. Proposed Method

- ➢ Log-tailored Language Model
- ➢ Contrastive Multi-modal Causal Structure Learning
- ➢ Causal Graph Fusion with KPI-Aware Attention

Experiment



III. Experiments

- ➢ Effectiveness

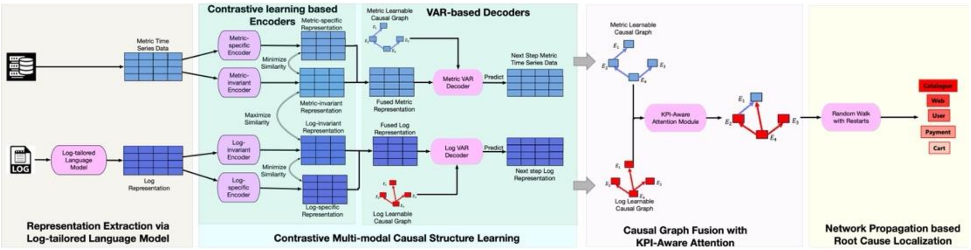
Conclusion

IV. Conclusion

Conclusion

❑Problem:

– Root Cause Analysis for microservice systems



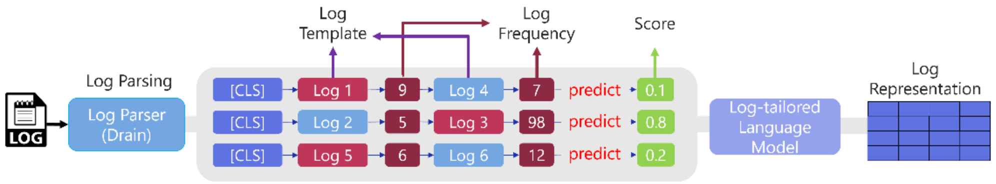
❑Algorithm (MULAN):

– Log-tailored Language Model

– Contrastive Multi-modal Causal Structure Learning

– Causal Graph Fusion with KPI-Aware Attention

– Network Propagation Based Root Cause Identification



Modality	Model	PR@1	PR@5	PR@10	MRR	MAP@3	MAP@5	MAP@10
Metric Only	Dynotears	0	0	0.50	0.070	0	0	0.075
	PC	0	0	0.25	0.053	0	0	0.050
	C-LSTM	0.25	0.75	0.75	0.474	0.5	0.25	0.675
	GOLEM	0	0	0.25	0.043	0	0	0.025
	REASON	0.75	**1.0**	**1.0**	0.875	0.917	0.95	0.975
Log Only	Dynotears	0	0	0.25	0.058	0	0	0.075
	PC	0	0	0.25	0.069	0	0	0.075
	C-LSTM	0	0	0.25	0.059	0	0	0.075
	GOLEM	0	0	0.25	0.058	0	0	0.075
	REASON	0	0.50	0.75	0.216	0.167	0.25	0.400
Multi-Modality	Dynotears	0	0	0.50	0.095	0	0	0.150
	PC	0	0	0.25	0.064	0	0	0.125
	C-LSTM	0.50	0.75	0.75	0.592	0.583	0.65	0.700
	GOLEM	0	0	0.25	0.065	0	0	0.050
	REASON	0.75	**1.0**	**1.0**	0.875	0.917	0.95	0.975
	Nezha	0	0.5	0.75	0.193	0.083	0.25	0.475
	MULAN	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

❑Experiments:

– Effectiveness evaluation on three real-world data sets.

Contact: Lecheng Zheng (Email: lecheng4@illinois.edu)

Contact: Lecheng Zheng (Email: lecheng4@illinois.edu)

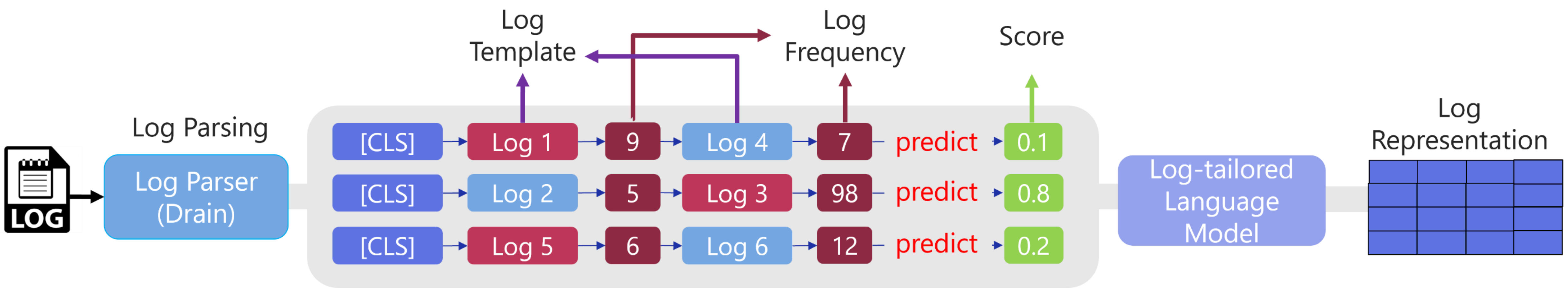
Log-tailored Language Model

❑Why is log frequency necessary?

– Illustrative example: DDoS attack

– In DDoS attack, the frequency of certain log templates may suddenly and dramatically increase, indicating unusual behavior.

– The frequency right after each log template provides extra information for monitoring unusual patterns in potential failure cases.



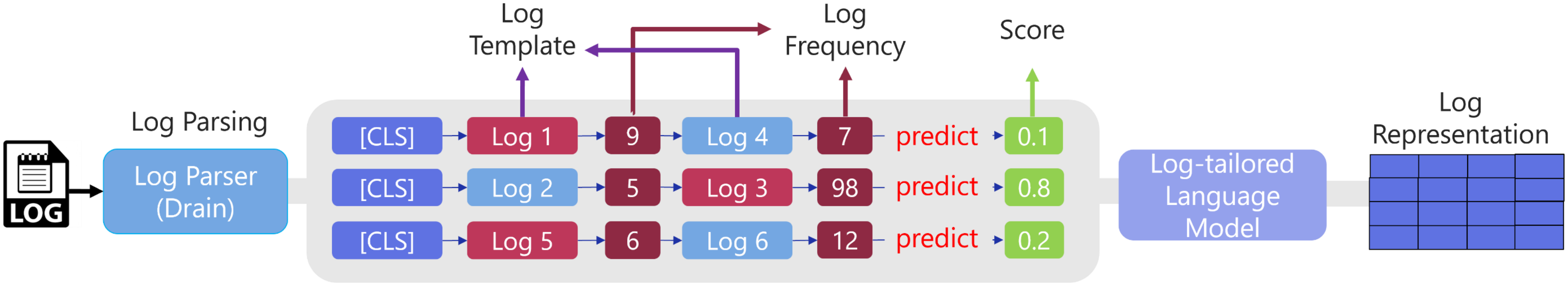
Log-tailored Language Model

- We leverage log-based anomaly detection algorithms (e.g., OC4Seq [5] or Deeplog [6]) to measure the anomaly score denoted as y^{log}.
- Objective Function

$$\mathcal{L}_{log} = \mathbb{E}_{i,j} \left\| y_i^{log} - f\left(X_{i,j}^L, c_{i,j}^L\right) \right\|^2$$

  – $c_{i,j}^L$ denotes a list of the frequency of the unique log templates within a log sequence $X_{i,j}^L$.

  – $f(\cdot)$ is the proposed language model that predicts the anomaly score.



- [5] Zhiwei Wang, et al. Multi-Scale One-Class Recurrent Neural Networks for Discrete Event Sequence Anomaly Detection. In SIGKDD 2021.
- [6] Min Du, et al. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. In SIGSAC 2017.

Overall Objective Function

- The final objective function is written as:

$$\mathcal{L} = \lambda_1 L_{var} + \lambda_2 L_{orth} + \lambda_3 L_{node} + \lambda_4 L_{edge} + \lambda_5 \big||A|\big|_1 + h(A)$$

  – $h(A) = tr(e^{A*A}) - n = 0$ if and only if A is acyclic.

  – $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ and λ_5 are the positive constant hyper-parameters.

Evaluation Metrics

- Precision@K (PR@K):

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

  – This metric measures the probability that the top-K predicted root causes are accurate.

- Mean Average Precision@K (MAP@K):

$$MAP@K = \frac{1}{K|\mathbb{A}|} \sum_{a \in \mathbb{A}} \sum_{i \le j \le K} PR@j$$

  – It provides an assessment of the top-K predicted causes from an overall perspective.

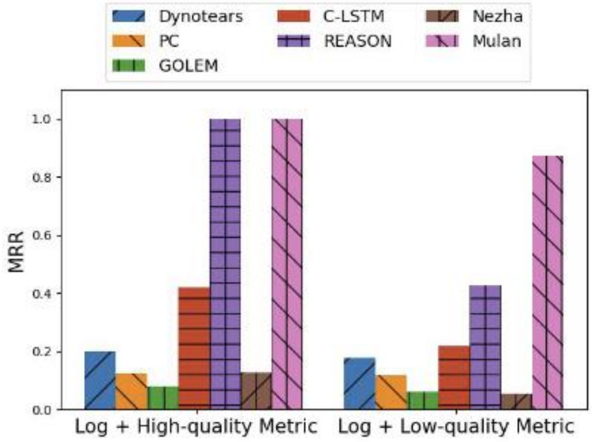
- Mean Reciprocal Rank (MRR):

$$MRR@K = \frac{1}{|\mathbb{A}|} \sum_{a \in \mathbb{A}} \frac{1}{rank_{R_a}}$$

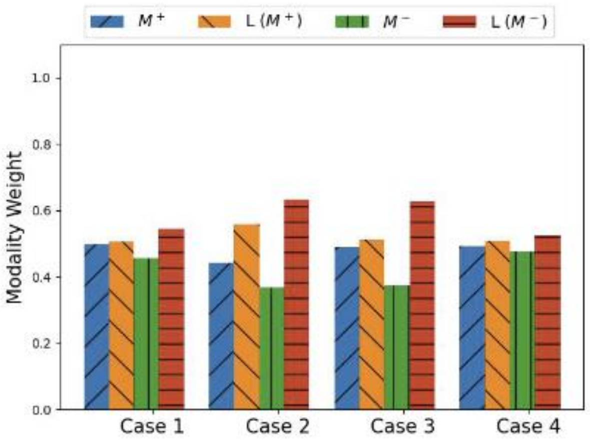
  – This metric evaluates the ranking capability of the models.

Case Study

- **Goal:** To demonstrate the robustness of our proposed method in the context of low-quality modality scenarios.
- Setup:
  - We assess the quality of distinct system metrics (e.g., CPU usage, memory usage, etc).
  - System metric with the highest median ranking score → the high-quality metric (M^+)
  - System metric with the lowest median ranking score → the low-quality metric (M^-)



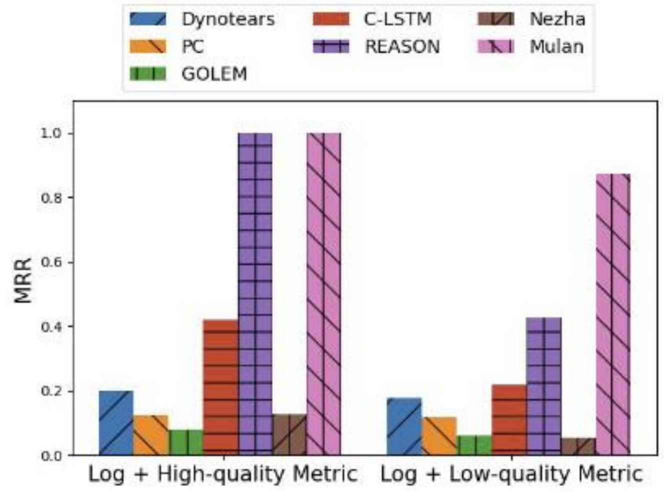
(a) Log + Metric



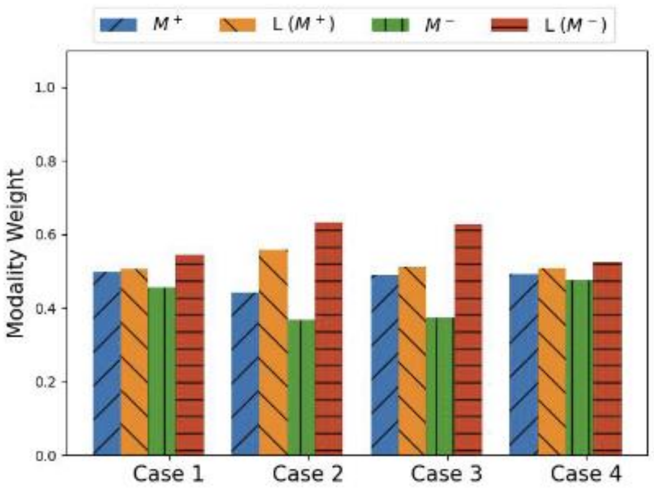
(b) Modality Weight

Case Study

- Figure (a): the performance undergoes a significant decline when the high-quality metric is substituted with the low-quality system metric.



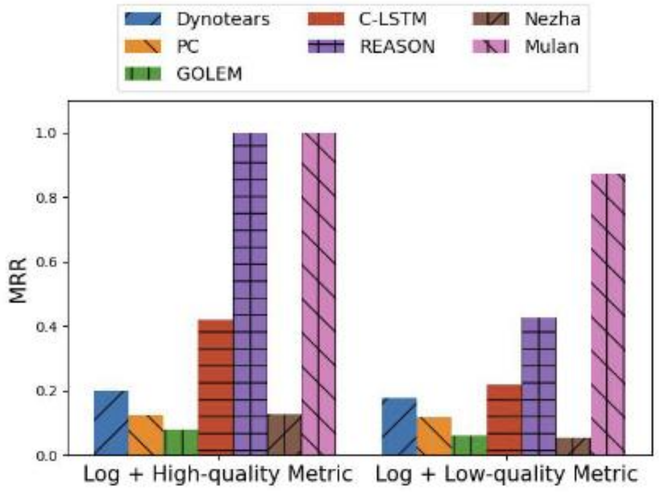
(a) Log + Metric



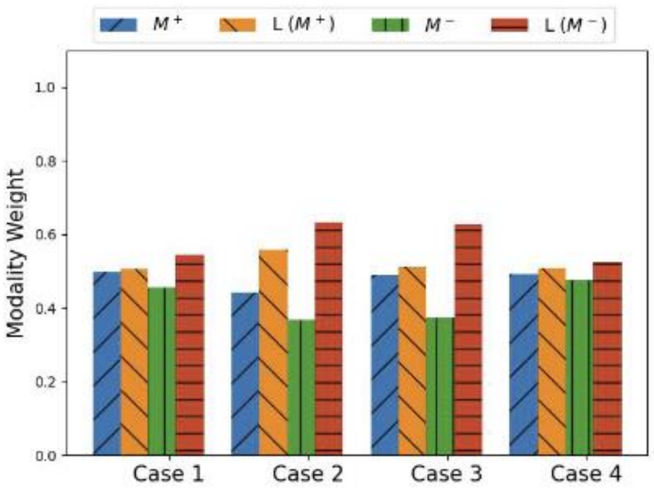
(b) Modality Weight

Case Study

- Figure (a): the performance undergoes a significant decline when the high-quality metric is substituted with the low-quality system metric.
- Figure (b): when the high-quality system metric (M^+ or blue bar) is replaced by the low-quality system metric (M^- or green bar), MULAN dynamically reduces the weight assigned to the system metric in all four cases.



(a) Log + Metric



(b) Modality Weight