Transactions on Computational and Scientific Methods | Vo. 4, No. 6, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Time-Varying Operational Graph Reasoning and Risk-Constrained Repair Planning for Tool-Augmented Large Language Model Agents

Yihang Tang
San Jose State University, San Jose, USA
erictang2019fall@gmail.com

Abstract: Addressing the operational challenges of cloud-native distributed systems, such as frequent topology reconfiguration, heterogeneous coupling of multi-source telemetry, and complex fault propagation paths, this paper studies a distributed automatic diagnosis and repair framework for operations and maintenance based on a large language model and tool-invoking intelligent agents. This framework uses a time-varying operations and maintenance graph as a unified semantic carrier, aligning and expressing observations such as service nodes, call relationships, and log indicator tracing within the same structure. It also introduces time-aware representation to enhance the characterization of non-stationary disturbances and change drift. Building upon this, the framework constructs a global representation for root cause localization through multimodal evidence fusion and context aggregation. An iterative belief update mechanism is employed to perform closed-loop absorption and correction of verifiable evidence returned by the tool, thus forming a continuous reasoning process from evidence retrieval and hypothesis generation to root cause inference. For automatic repair requirements, the framework models action selection as a planning problem that considers utility, cost, and risk constraints. Risk budgeting and rollback conditions are used to constrain the repair sequence, ensuring the controllability and auditability of the output actions. Comparative experiments on public benchmarks validated the advantages of the proposed framework in root cause localization, ranking, and remediation effectiveness, demonstrating that the method can achieve more reliable automatic diagnosis and remediation decisions in complex operational data and toolchain environments.

Keywords: Cloud-native operations and maintenance; root cause analysis; closed-loop reasoning; risk constraint planning.

1. Introduction

The widespread adoption of cloud computing and microservice architectures has driven the continuous growth of computing resources and business traffic[1]. Distributed systems, in their large-scale evolution, exhibit greater heterogeneity and dynamism. Frequent refactoring of service dependencies, routine version iterations and configuration changes, and cross-regional deployments and multi-cluster collaboration bring more complex spatiotemporal coupling effects. Simultaneously, operational data such as logs, metrics, and tracing data exhibit high-dimensionality, multi-source characteristics, strong noise, and strong correlations. Alarm storms and false alarm/missed alarm issues persist, significantly lengthening the fault location and handling process[2]. Traditional rule-based operational methods lack robustness in complex scenarios such as

time-varying dependency topologies, shifting load patterns, and cascading propagation, making it difficult to complete high-quality root cause analysis and recovery decisions within a limited time window.

Automatic diagnosis and repair in distributed operations and maintenance have significant practical needs and engineering value. Faults often appear early as weak signals and spread along the call chain and shared resources, forming a chain reaction of degradation across services and components, leading to performance fluctuations, decreased availability, and business losses[3]. Operational and maintenance (O&M) processes typically involve collaborative efforts across multiple stages, including log retrieval, backtracking, change alignment, configuration verification, capacity assessment, and rollback or rerouting. These processes rely heavily on expert experience and the combined use of multiple tools, resulting in complex workflows with extremely high timeliness requirements. In high-frequency change and high-concurrency environments, a single-point perspective and a single data source are insufficient for accurate decision-making. There is an urgent need for intelligent frameworks capable of integrating multi-source evidence, understanding system semantics, and generating executable operation sequences to reduce mean time to recovery and improve system resilience[4].

Large language models demonstrate excellent adaptability to unstructured and semi-structured O&M knowledge in semantic understanding, reasoning, and planning, providing a new technical path for automating complex O&M processes. In O&M scenarios, a large amount of knowledge is stored in textual forms such as work order records, fault debriefings, operation manuals, and configuration instructions, while also being closely linked to time-series indicators, topology relationships, and event sequences[5,6]. By combining large language models with tool-invoking capabilities, the diagnostic process can be extended from pure text generation to a closed-loop task solution based on evidence retrieval, analysis, and verification. This allows the model to select appropriate observation and operation tools through multiple rounds of reasoning, gradually converging to interpretable root cause hypotheses and remediation strategies. Compared to static knowledge question answering, agents with tool-invoking capabilities can execute queries and verifications in real-world operational environments and dynamically adjust decision paths based on real-time feedback, thereby improving diagnostic reliability and remediation controllability.

Research on automated diagnostic and remediation frameworks for distributed operations and maintenance has significant theoretical and practical implications. Theoretically, it is necessary to construct a unified representation and reasoning mechanism for multi-source operational data, addressing key issues such as evidence consistency modeling, causal chain constraints, state space incompleteness, and uncertainty propagation, and developing verifiable task planning and action selection methods. In application, it is necessary to achieve secure orchestration and access control for complex toolchains, ensuring that remediation actions have rollback capabilities and risk constraints, avoiding secondary failures and amplification effects caused by automation[7]. By constructing a distributed operation and maintenance automatic diagnosis and repair framework based on a large language model and tool-calling intelligent agents, the efficiency of fault handling and system stability can be improved while ensuring safety and controllability, providing support for the highly reliable operation of large-scale cloud services and critical business systems.

2. Methods

Facing the challenges of frequently changing dependency topologies and heterogeneous coupling of multi-source evidence in distributed operations and maintenance scenarios, the framework abstracts the runtime state as a time-varying O&M graph to uniformly host service relations and observational information, thereby providing a structured semantic foundation for subsequent diagnostic reasoning and repair planning. This paper presents the overall model architecture, as shown in Figure 1.

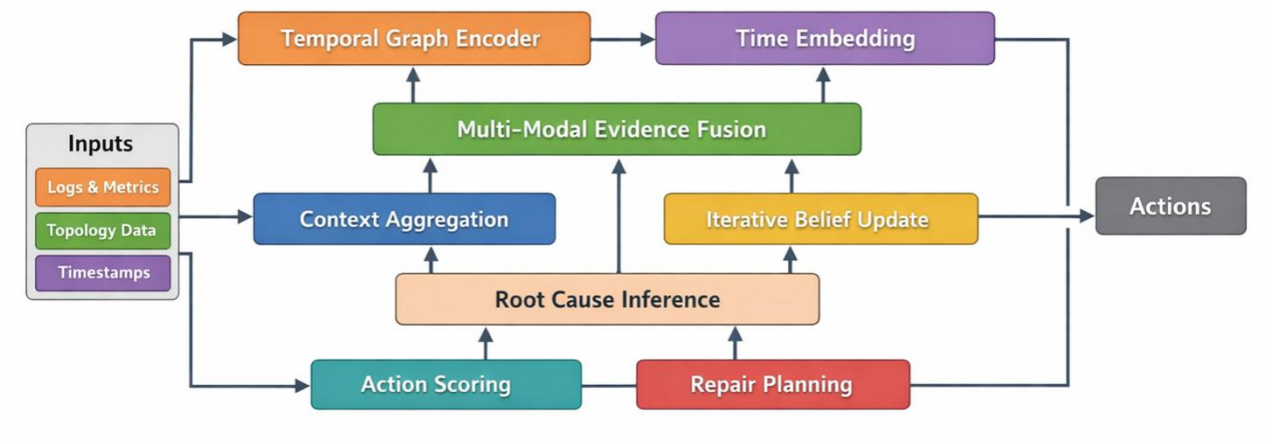

Figure 1. Overall model architecture

The system representation at time t is defined as a graph structure containing a node set $\mathcal{V}_t$ and an edge set $\mathcal{E}_t$, while the node feature matrix $\boldsymbol{X}_t$ aggregates observations such as metric statistics and log event counts, and the edge feature matrix $\boldsymbol{R}_t$ characterizes link attributes such as call latency distributions and error-code ratios, so that topology and data are coupled within the same object.

$$\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \boldsymbol{X}_t, \boldsymbol{R}_t)$$

To strengthen the characterization of non-stationary temporal patterns and change-induced drift, the timestamp τ is mapped to an embedding vector $\boldsymbol{\phi}(\tau)$ where periodicity and trend coexist, in which the linear component $w_0\tau + b_0$ expresses slow trends, while several sinusoidal components $\sin(w_i\tau + b_i)$ capture periodic perturbations, thereby providing distinguishable time-phase information for subsequent representations.

$$\boldsymbol{\phi}(\tau) = [w_0\tau + b_0, sin(w_1\tau + b_1), \ldots, sin(w_m\tau + b_m)]^{\top}$$

To address the instability of a single view caused by noise and missing values in multi-source O&M signals, the encoding stage fuses local evidence of nodes and edges on the graph backbone, enabling service-level representations to absorb information such as metric intensity, log semantics, and link statistics. The hidden vector $\boldsymbol{h}_v$ of node v is obtained by linearly transforming the node observation vector $\boldsymbol{x}_v$, the relation vector $\mathbf{r}_v$ aggregated from its adjacent links, and the time embedding $\boldsymbol{\phi}(\tau)$, followed by a non-linear function $\sigma(\cdot)$, thereby achieving cross-modal alignment and temporal-awareness injection within the same representation space.

$$\boldsymbol{h}_v = \sigma\left(\boldsymbol{W}_x\boldsymbol{x}_v + \boldsymbol{W}_r\boldsymbol{r}_v + \boldsymbol{W}_t\boldsymbol{\phi}(\tau)\right)$$

To highlight critical evidence during global diagnosis and suppress redundancy caused by alert storms, the framework adopts correlation-based weighted aggregation to obtain a global context vector $\boldsymbol{z}$, where the weight α_i is determined by the matching degree between the query vector $\boldsymbol{q}$ and the candidate evidence representation $\boldsymbol{h}_i$, so that highly relevant evidence occupies a larger proportion in aggregation and improves the interpretability and stability of root-cause inference.

$$\alpha_i = \frac{exp\left(\boldsymbol{q}^\top \boldsymbol{h}_i\right)}{\sum_j \ exp\left(\boldsymbol{q}^\top \boldsymbol{h}_j\right)}$$

$$\boldsymbol{z} = \sum_i \alpha_i \boldsymbol{h}_i$$

Considering that the diagnostic process is essentially a closed-loop reasoning procedure in which evidence is accumulated step by step, and hypotheses are continuously revised, the agent maintains a belief state $\boldsymbol{b}_k$ that evolves with step k and updates it after each tool observation, so that the reasoning chain can be constrained by real-time feedback and avoid the hallucination risk brought by one-shot generation. The observation result $\boldsymbol{o}_k$ is returned by retrieval and monitoring tools, the fusion vector $\boldsymbol{z}_k$ provides cross-modal context, the update function $f\left(\boldsymbol{z}_k, \boldsymbol{o}_k\right)$ outputs incremental information aligned with the candidate root-cause space, and the step-size coefficient $\eta \in (0,1)$ controls the influence strength of new evidence on historical belief to improve robustness.

$$\boldsymbol{b}_{k+1} = (1-\eta)\boldsymbol{b}_k + \eta f\left(\boldsymbol{z}_k, \boldsymbol{o}_k\right)$$

To meet the requirement that automated repair should balance effectiveness and safety, action selection is modeled as a scoring problem that trades off benefits and costs, enabling the agent to consider recovery objectives and risk constraints during planning. The score $s(a)$ of a candidate action a is determined jointly by the conditional utility $u(a|\boldsymbol{b})$, the execution cost $c(a)$, and the risk function $r(a)$, where hyperparameters λ and μ control the penalty strengths for resource consumption and potential side effects, respectively, thereby suppressing aggressive operations and encouraging conservative repair sequences with rollback capability.

$$s(a) = u(a|\boldsymbol{b}) - \lambda c(a) - \mu r(a)$$

Furthermore, repair is organized as an action sequence $a_{1:K}$ of length K, and the planning objective maximizes the cumulative score subject to a total risk budget $\mathcal{R}_{\max}$, while using the constraint indicator function $g\left(a_k\right) = 1$ to express that an action has a rollback path or reversible conditions, thereby explicitly injecting controllability into the decision layer and reducing the probability of secondary failures.

$$a^*_{1:K} = \underset{a_{1:K}}{argmax} \sum_{k=1}^{K} s\left(a_k\right)$$

$$\sum_{k=1}^{K} r\left(a_k\right) \le \mathcal{R}_{max}, \ g\left(a_k\right) = 1$$

3. Experimental Results and Analysis

3.1 Dataset

This paper selects the RE2 Train Ticket dataset from the RCAEval benchmark as the research dataset. This dataset is designed for root cause localization and diagnosis tasks in microservice distributed operations and maintenance (O&M). It covers various fault injection types and provides labeled root cause services and root cause indication signals for each fault sample, thus supporting complete process modeling from evidence

aggregation to diagnostic conclusion generation. The dataset organizes fault cases around microservice systems, emphasizing the challenges of localization under cross-service propagation and multi-component coupling, which aligns with the topological correlation and causal chain inference requirements of automated diagnosis and repair frameworks for distributed O&M. In addition, repair interventions are simulated via a counterfactual mechanism that perturbs candidate root-cause services and re-evaluates telemetry changes within the same fault window, thereby approximating remediation effects without executing real actions in production.

The data content is centered on multi-source telemetry, including three types of observations: indicators, logs, and link tracing. Time series segments from different sources are aligned using a unified time field, enabling consistent retrieval, comparison, and fusion of multi-source evidence within the same fault window. In addition to the multi-source data itself, the dataset also provides fault type-based organization and root cause service-level annotation information, facilitating the representation of the diagnostic process as inference and ranking in the candidate root cause space. It also provides actionable monitoring signals and evaluation references for subsequent tool-based evidence verification and action planning.

3.2 Experimental setup

The experimental setup of this study is summarized in Table 1. The configuration focuses on the hardware and software operating environment and key training hyperparameters to ensure that the distributed operation and maintenance, automatic diagnosis and repair framework can be reproduced under unified computing resources and consistent randomness control. The basic large model uses Qwen7B, the software stack is mainly Linux and PyTorch, and AdamW optimization and gradient pruning are used on the training end to stabilize updates. Unnecessary fluctuations are reduced by using a fixed random seed and deterministic settings, which facilitates the alignment and comparison of each module of the model under the same settings.

Table 1: Hyperparameter settings

Item	Setting
Base LLM	Qwen7B
OS	Ubuntu 20.04 LTS
CPU	Intel Xeon 32 vCPU
GPU	NVIDIA RTX 3090 24GB
Memory	128 GB
Deep learning framework	PyTorch 2.3
Python	3.10
CUDA	11.8
Decoding strategy	Greedy / temperature 0.2
Max output length	1024
Tool calling	Function-call style, JSON Schema constrained
Optimizer	AdamW
Learning rate	1e-3
Weight decay	1e-4
Batch size	64
Training epochs	50

3.3 Experimental Results and Analysis

To verify the effectiveness of the distributed operation and maintenance automatic diagnosis and repair framework in key capabilities such as multi-source telemetry evidence fusion, root cause localization, and controllable repair planning, this section selects publicly available research works consistent with this research direction as comparative objects and provides a horizontal comparison under a unified evaluation index. The comparative works cover representative routes such as microservice root cause analysis, tool-assisted agent reasoning, and event response and automated handling for operation and maintenance scenarios, thereby supporting a comprehensive evaluation of diagnostic accuracy, sorting quality, and repair execution effectiveness. The experimental results are shown in Table 2.

Table 2: Comparison results of related methods

Method	RCA Top-1 Acc	RCA Top-3 Acc	MRR	Repair Success Rate
Chen et al.[8]	0.62	0.79	0.68	0.54
Zhang et al.[9]	0.65	0.82	0.70	0.56
Li et al.[10]	0.60	0.78	0.66	0.52
Zhang et al.[11]	0.67	0.84	0.72	0.58
Yao et al.[12]	0.63	0.80	0.69	0.55
Pham et al.[13]	0.61	0.77	0.67	0.53
Zhang et al.[14]	0.66	0.83	0.71	0.57
Ours	0.72	0.88	0.76	0.63

The results shown in Table 2 demonstrate that the proposed framework exhibits stronger overall performance in root cause localization, ranking, and remediation effectiveness. By semantically aligning and aggregating multi-source telemetry evidence within a unified structure and continuously absorbing verification information returned by the tool during closed-loop inference, a more stable chain of evidence and more consistent root cause determination can be formed during the diagnostic phase, thus maintaining higher reliability and interpretability across different evaluation dimensions. This advantage stems not only from the joint modeling of log indicator chains but also from the ability of time-aware representations to characterize non-stationary disturbances, making key anomaly clues easier to highlight and use to support decision-making in complex noise backgrounds.

The remediation performance reflects the strengthening effect of action selection and planning mechanisms on security and controllability. By simultaneously incorporating utility costs and risk constraints into the scoring function and taking rollback as a prerequisite for execution, the remediation strategy can reduce unnecessary side effects while meeting recovery objectives, thereby improving the overall availability and stability of the handling. Meanwhile, the iterative update of belief states keeps the planning process sensitive to new evidence, avoiding overly aggressive operations when information is incomplete, thus making the final automatic remediation more in line with the core requirements of robustness and controllability in distributed operation and maintenance scenarios.

To characterize the dependence of the optimization process on key training hyperparameters, this paper conducts a learning rate sensitivity analysis to examine the stability of diagnostic capabilities under different

update steps. This setting can reveal the mechanism by which changes in gradient update magnitude affect model convergence behavior and generalization performance. By switching learning rate configurations under consistent data partitioning and training strategies, an interpretable basis can be provided for hyperparameter selection during engineering deployment. The experimental results are shown in Figure 2.

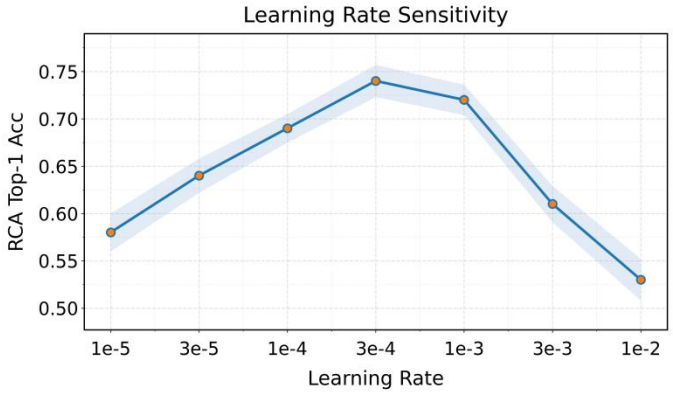

Figure 2. Sensitivity experiment of learning rate to RCA Top-1 Acc

Sensitivity analysis shows that the proposed method maintains strong root cause localization capability when the learning rate is set reasonably, indicating good adaptability of the training process to the update step size. At a moderate learning rate, parameter updates can fully absorb effective information from multi-source evidence without disturbing semantic alignment and evidence aggregation due to excessively large step sizes, thus making the diagnostic output more stable and reliable. This phenomenon is related to the synergy between time-aware representation and evidence fusion mechanisms in the framework, making it easier for key anomaly cues to be continuously reinforced during training and form consistent root cause discrimination criteria.

When the learning rate deviates from the effective range, the root cause localization capability is affected, reflecting the requirement for optimization stability based on the quality of representations upon which closed-loop reasoning and repair decisions depend. Too small a step size reduces the efficiency of effective information accumulation, making it difficult for belief state updates to fully absorb tool feedback and structural evidence, while too large a step size may introduce oscillations and over-updates, weakening the consistency of the evidence chain and affecting the reliability of action scoring and planning. Overall, the proposed method performs better with an appropriate learning rate configuration and provides a clear reference for hyperparameter selection in the engineering deployment phase.

4. Conclusion

This paper addresses the pain points of complex fault modes, diverse evidence sources, and long, experience-dependent handling processes in distributed operations and maintenance (O&M) scenarios. It proposes an automated diagnosis and repair framework based on a large language model and tool-invoking intelligent agents. This framework uses a time-varying O&M graph to uniformly represent topological relationships and multi-source telemetry information. It enhances the characterization of non-stationary disturbances and cascading propagation through time-aware representation and cross-modal evidence fusion. During closed-loop reasoning, it continuously supplements verifiable evidence using tool invocation, forming an end-to-end closed-loop process from root cause inference to action decision-making. The overall design emphasizes the coupling of semantic understanding and executable operations, enabling diagnosis to move beyond textual interpretation and onto an auditable and constrained handling chain, providing a more engineering-feasible intelligent path for the reliable operation of complex systems.

At the methodological level, the framework highlights the crucial role of controllability and security constraints in automated repair. By modeling repair as a sequential decision-making process that weighs

utility, cost, and risk, and explicitly integrating constraints such as rollbackability and risk budgeting into the action selection and planning mechanisms, the handling strategy can suppress unnecessary side effects while meeting recovery objectives. Meanwhile, the iterative updating of belief states enables the reasoning process to be continuously corrected through real-time observation and tool feedback, thereby reducing the probability of misjudgment under incomplete information conditions and improving decision consistency. This paradigm combines the reasoning and planning capabilities of large language models with the verifiability of operation and maintenance toolchains, providing general methodological support for building more reliable autonomous operation and maintenance systems.

From an application impact perspective, the proposed framework has direct value for cloud computing platforms, microservice business systems, and multi-cluster hybrid deployment environments. Its multi-source evidence fusion capability can alleviate the difficulty of localization caused by alarm storms and noise interference. The tool-invoked closed-loop verification mechanism helps shorten the path from anomaly discovery to root cause confirmation, and improves handling efficiency and stability through controllable repair strategies. For scenarios requiring high availability, such as online services, industrial internet, and critical infrastructure, this framework is expected to lower the operation and maintenance threshold and improve system resilience, promoting the evolution from experience-centered manual response to process-centered intelligent autonomy. It also provides a structured carrier for the accumulation and reuse of operation and maintenance knowledge, driving operation and maintenance practices towards greater standardization and auditability.

Looking to the future, research can continue to advance within a framework of enhanced interpretability and stricter security governance. On the one hand, it can further strengthen the traceability of evidence chains and causal constraints, enabling root cause identification and remediation decisions to be presented with clearer reasoning trajectories and facilitating auditing and review. On the other hand, it can expand the unified orchestration mechanism with interfaces to more operational systems, introducing finer-grained permission isolation, change verification, and rollback orchestration strategies to improve consistency and risk control capabilities across domains. Simultaneously, combining the framework with continuous learning mechanisms to adapt to version iterations and topology refactoring, and exploring privacy protection and policy sharing in multi-tenant and multi-organizational environments, will help further expand its application boundaries and practical impact in large-scale cloud-native operations and intelligent reliability engineering.

References

[1] Ikram A, Chakraborty S, Mitra S, et al. Root cause analysis of failures in microservices through causal discovery[J]. Advances in Neural Information Processing Systems, 2022, 35: 31158-31170.

[2] Sarda K, Namrud Z, Rouf R, et al. Adarma auto-detection and auto-remediation of microservice anomalies by leveraging large language models[C]//Proceedings of the 33rd Annual International Conference on Computer Science and Software Engineering. 2023: 200-205.

[3] Las-Casas P, Kumbhare A G, Fonseca R, et al. LLexus: an AI agent system for incident management[J]. ACM SIGOPS Operating Systems Review, 2024, 58(1): 23-36.

[4] Zha J, Shan X, Lu J, et al. Leveraging large language models for efficient alert aggregation in aiops[J]. Electronics, 2024, 13(22): 4425.

[5] Yu Z, Ouyang Q, Pei C, et al. Causality enhanced graph representation learning for alert-based root cause analysis[C]//2024 IEEE 24th International Symposium on Cluster, Cloud and Internet Computing (CCGrid). IEEE, 2024: 77-86.

[6] M. Soldani and A. A. Salah, "Explainable Artificial Intelligence for IT Operations (AIOps): Opportunities and Challenges," IEEE Access, vol. 12, pp. 39515-39534, 2024.

[7] J. Bogatinovski, S. Nedelkoski, J. Cardoso, and O. Kao, "AIOps: Real-World Challenges and Research Innovations," in Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS), 2020, pp. 1-9.

[8] C. S. Xia, Y. Deng, S. Dunn, and L. Zhang, "Agentless: Demystifying LLM-based Software Engineering Agents," in Proc. ACM SIGSOFT FSE, 2024.

[9] S. Wang, A. Zhou, M. Zaheer, et al., "GraphRCA: Efficient Root Cause Analysis for Large-Scale Microservice Systems," in Proc. IEEE ICDCS, 2023.

[10] Roy D, Zhang X, Bhave R, et al. Exploring llm-based agents for root cause analysis[C]//Companion proceedings of the 32nd ACM international conference on the foundations of software engineering. 2024: 208-219.

[11] C. S. Xia, Y. Deng, S. Dunn, et al., "Automated Program Repair in the Era of Large Pre-trained Language Models," ACM Computing Surveys, vol. 56, no. 8, 2024.

[12] Yao Z, Pei C, Chen W, et al. Chain-of-event: Interpretable root cause analysis for microservices through automatically learning weighted event causal graph[C]//Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering. 2024: 50-61.

[13] Y. Gan, M. Liang, S. Dev, et al., "Sage: Practical and Scalable ML-Driven Performance Debugging in Microservices," in Proc. ASPLOS, 2019, pp. 699-714.

[14] P. Chen, Y. Qi, D. Hou, et al., "MicroRank: End-to-End Latency Issue Localization with Extended Spectrum Analysis in Microservice Environments," in Proc. WWW, 2021, pp. 3087-3098.