Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 3, No. 7, 2024

Learning Unified Multi-Granularity Representations for Backend Anomaly Detection and Causal Localization

Yun Yang[1], Chenfeiyu Wen[2], Hengguang Cui[3],Yilin Sun[4], Yibo Liu[5*]

[1]Northeastern University, Boston, USA

[2]New York University, New York, USA

[3]Brown University, Providence, USA

[4]University of Pennsylvania, Philadelphia, USA

[5]Carnegie Mellon University, Pittsburgh, USA*

*Corresponding Author: Yibo Liu, lyb8812@outlook.com

Abstract: As backend systems grow in scale and architecture complexity, performance anomalies often exhibit characteristics such as multi-source signal coupling, cross-component propagation, and varied morphologies, making it difficult for solutions relying solely on single observations or single-granularity modeling to balance accuracy and interpretability. This paper proposes a unified framework based on self-supervised representation learning for multi-granularity backend performance anomaly detection and root cause inference. This framework aligns and normalizes heterogeneous telemetry data such as metrics, logs, and tracking within the same time window, and learns stable latent representations through a shared encoder. In terms of method design, the framework simultaneously characterizes temporal context and dependency structure information: on the one hand, it extracts window-level states using temporal aggregation; on the other hand, it constructs a graph structure based on call relationships and performs relational representation fusion. Subsequently, a self-supervised objective with consistency constraints is adopted to improve the robustness of the representation to noise and scene fluctuations. In the inference stage, this paper constructs an anomaly scoring mechanism in the latent space and combines neighborhood consistency for score propagation and aggregation, thereby outputting root cause ranking results that can be used for localization. Through systematic comparison on public benchmarks, the proposed method demonstrates stronger overall advantages in both detection quality and root cause hit capability, verifying the effectiveness and practical value of multi-source telemetry alignment, multi-granularity representation learning, and structural consistency inference in backend anomaly diagnosis tasks.

Keywords: Backend telemetry fusion, self-supervised alignment, multi-granularity representation, root cause ranking

1. Introduction

With the widespread adoption of cloud-native and microservice architectures, modern backend systems are characterized by a large number of components, long call chains, and frequent dynamic changes in the operating environment[1]. Sudden fluctuations in business traffic, behavioral drift caused by version iterations, and cascading effects triggered by resource contention make performance anomalies more insidious and sudden. Traditional methods relying on threshold rules or single-metric alerts often struggle to cover complex scenarios, leading to false positives and false negatives, and losing their interpretability when propagating across services and layers. Therefore, performance anomaly detection and root cause inference for backend systems are crucial not only for fault location efficiency and system availability but also for service stability, user experience, and operational costs[2].

In actual operation, backend performance anomalies typically exhibit a mixed characteristic across multiple forms and scales[3]. For example, phenomena such as latency tail lifting, decreased throughput, increased error rates, queue backlogs, or resource saturation may occur concurrently and propagate along the topology at different granularities. Modeling anomalies from a single perspective often fails to account for both local fine-grained fluctuations and global cross-component correlations, resulting in insensitivity to early anomaly signals and difficulty in characterizing complex failure modes. A multi-granular perspective can unify the organization of observational information at the host, container, service, interface, and call chain levels, enabling models to capture both short-term changes in micro-level perturbations and structural shifts in macro-level behavior. This provides a more solid foundation for interpretable anomaly detection and root cause localization.

Meanwhile, backend observability data is typically characterized by sparse supervision and ambiguous semantics. In real production environments, anomalies are relatively rare events; their occurrences are highly uneven across services and time, and their manifestations can vary widely even when the underlying cause is similar. This makes it difficult to obtain

representative labeled samples, and labels themselves can be noisy because incident boundaries, symptom severity, and remediation actions are often recorded with operational rather than modeling objectives. In addition, what appears abnormal in one workload phase may be normal in another, and the same metric fluctuation may correspond to very different system states depending on the surrounding context. As a result, purely label-driven learning can become brittle, costly to maintain, and prone to blind spots when the system evolves or when novel failure modes appear.

Self-supervised representation learning offers a practical and principled alternative in this setting. By leveraging abundant unlabeled runtime data, it can learn stable and transferable representations via carefully designed pretraining objectives that emphasize consistency under perturbations and context-aware structure. This allows the model to absorb measurement noise, workload drift, and service heterogeneity into the representation space, while still preserving discriminative factors that separate abnormal patterns from normal variability. More importantly, self-supervised learning can turn high-dimensional, multi-source, and asynchronous telemetry streams into alignable latent structures, enabling cross-time, cross-component, and cross-granularity comparisons under a unified semantic framework, which is critical for reliable anomaly detection and subsequent root cause inference in complex backend systems[4].

From a systems governance perspective, if anomaly detection cannot be linked to root cause inference, it's difficult to form a closed-loop decision-making process. Alerts can only detect problems rather than explain them and guide remediation. Root cause inference needs to answer key questions simultaneously, such as where the anomaly originated, through what path it spreads, and which components are most likely to trigger performance degradation. It must consider both temporal causal clues and system topology and dependencies. Multi-granularity methods based on self-supervised representation learning promise to unify detection and inference within the same representation and reasoning framework. This improves the robustness of anomaly identification while enhancing interpretability and operability, thereby supporting automated operation and maintenance, rapid loss mitigation, and continuous optimization, providing crucial support for the long-term stable operation of highly reliable backend systems.

2. BackGround

In large-scale backend scenarios, the very methods of generating and collecting observational data complicate anomaly understanding. Metrics, logs, and tracing data often differ in time granularity, sampling frequency, and statistical caliber, and data frequently contains missing data, jitter, and asynchronous alignment biases[5]. Furthermore, system operation is affected by factors such as capacity scaling, scheduling migration, cache warming, fluctuations in dependent services, and multi-tenant resource contention, leading to significant phased and context-dependent behavior in normal operations[6]. A direct problem arising from this is that many seemingly anomalous signals may simply be natural results of workload switching or policy adjustments, while genuine performance degradation may occur scattered among multiple weak signals, increasing the threshold for judgment and interpretation.

To address this complexity, existing methods can be broadly categorized by modeling object and information utilization method, including rule-based monitoring and alerting, statistical and time-series-based anomaly scoring, topology-based correlation analysis, and end-to-end modeling based on representation learning. These different approaches differ significantly in data dependency, interpretability, portability, and maintenance costs, directly impacting their effectiveness in production environments[7]. To facilitate the clarification of problem boundaries and technical entry points in subsequent research, this paper summarizes the key features of related approaches in Table 1 and uses this as the background for proposing new methods.

Table 1: Feature comparison of different backend anomaly analysis routes

Route category	Main inputs	Representative ideas	Strengths	Limitations
Rule- and threshold-based alerting	Single metric or a small set of metrics	Static thresholds; combinations of alerting policies	Fast to deploy; low cost; directly actionable for operations	Sensitive to business fluctuations; hard to cover complex patterns; high policy maintenance burden
Statistical and time-series detection	Metric time series	Change-point detection; forecasting residuals; distribution shift	Sensitive to short-term spikes; suitable for stable metrics	Relies on consistent definitions/telemetry; assumes stationarity; weak cross-component coupling
Correlation and topology analysis	Metrics plus dependency relations	Propagation paths; association strength; structural constraints	Helps identify impact scope and upstream/downstream relations	Sensitive to topology accuracy and data completeness; hard to handle implicit dependencies
Representation learning methods	Multi-source observational data	Representation alignment; anomaly scoring; similarity retrieval	Adapts to multimodal inputs and complex patterns; offers some generalization potential	Sensitive to training objective design; interpretability requires additional mechanisms
Root-cause inference methods	Observations plus structural information	Attribution, ranking; fusion of causal cues	Decision-oriented; outputs are more actionable for remediation	Vulnerable to confounders; challenges in the stability and consistency of inference

3. Method

This paper's method uses unlabeled runtime data as the core input. First, it unifies backend observations into learnable sequence and graph structure representations. Then, it learns robust representations through self-supervised objective learning. Finally, it completes anomaly scoring and root cause ranking in a multi-granularity space. Let $X = \{x_t\}_{t=1}^{T}$ be the multi-source observations collected within the time window, where $x_t \in R^d$ is composed of various indicators and derived statistics. To reduce interference from different dimensions and noise, dimensionality standardization and smoothing are performed first, ensuring comparability of the same service across different load stages. Then, a shared encoder f_θ maps each time step to a latent space $z_t \in R^m$, and representations are constructed simultaneously at the service and call relationship levels, forming a multi-granularity embedding set $Z = \{z_{s,t}, z_{c,t}\}$. Here, $z_{s,t}$ represents the state representation at the service granularity, and $z_{c,t}$ represents the interaction representation at the call edge or link segment granularity. The significance of multi-granularity lies in incorporating local minor fluctuations and global structural shifts into the same metric space, enabling subsequent anomaly detection to utilize both single-point changes and propagation consistency and dependency constraints. Its overall model architecture is shown in Figure 1.

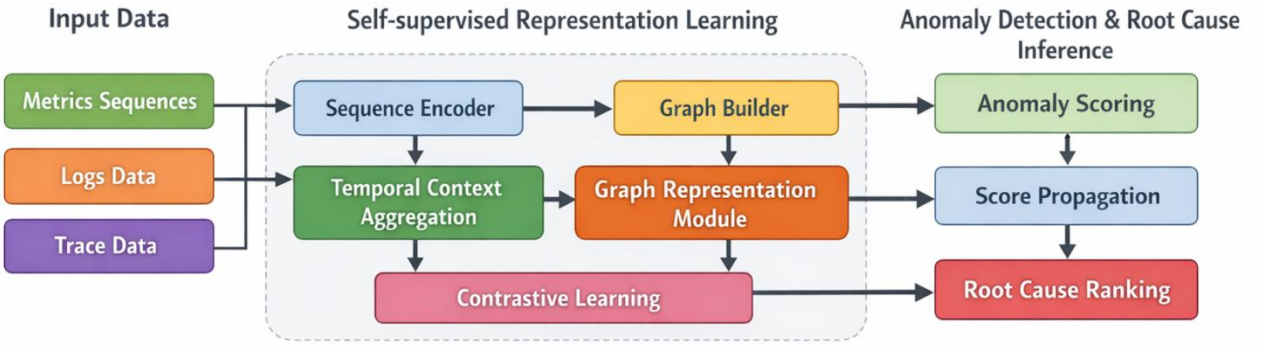

Figure 1. Overview of the proposed multi-granularity backend performance anomaly detection and root cause inference framework based on self-supervised representation learning. Multi-source observability signals are encoded into temporal and graph representations, trained via contrastive learning, and then used for anomaly scoring, score propagation, and root cause ranking.

The encoding phase employs a lightweight combination of temporal and relational aggregation. On the temporal side, a sliding aggregation operator g is used to extract the context within a window, for example, obtaining a smooth state h_t through first-order recursive updates. On the relational side, service dependencies are abstracted as a directed graph $G = (V, E)$, where each node corresponds to a service or component, each edge corresponds to a call dependency, and normalized weighted aggregation is performed on the neighbor representations. The core mapping relationship can be expressed in the following simple form:

$$z_t = f_\theta(x_t)$$

$$h_t = \alpha h_{t-1} + (1-\alpha) z_t$$

$\alpha \in [0,1)$ controls the trade-off between short-term and long-term information, while h_t serves as a window-level representation for subsequent comparative learning and anomaly scoring.

The self-supervised learning part employs the idea of consistency alignment. It obtains two views by applying two light perturbations to the same window, requiring their latent representations to remain close, thus learning representations insensitive to noise and random jitter under unlabeled conditions. Let u and v be the two views encoded from the same window. Then, a simple cosine similarity is used as the consistency measure, and the loss is constructed based on this.

$$sim(u,v) = \frac{u^T v}{\|u\|\|v\|}$$

$$L_{ssl} = 1 - sim(u,v)$$

This objective tends to compress the diverse variations of normal operation onto a relatively stable manifold, making it easier to distinguish true structural shifts in the latent space, while also providing a unified similarity basis for cross-service and cross-granularity alignment.

Anomaly detection and root cause inference are performed within the same representation space. First, time-level anomaly scores are given in the form of predicted or reconstructed residuals. Then, consistency propagation is performed on the graph structure to form an interpretable root cause ranking. Taking the simplest linear predictor ϕ as an example, $\hat{z}_{t+1} = \phi(h_t)$ can be used to generate the predicted embedding for the next time step, and the anomaly strength is defined by the residual norm.

$$s_t = \|z_{t+1} - \hat{z}_{t+1}\|_2$$

At the root cause level, the multi-granularity scores of node i are aggregated within a window to obtain r_i, and then combined with its in-degree or influence range to form the final ranking score q_i. A simple ranking method is to weightily combine the node score with its neighbor scores:

$$q_i = \beta r_i + (1-\beta)\frac{1}{|N(i)|}\sum_{j\in N(i)} r_j$$

Where $N(i)$ is the set of neighbors of node i, and $\beta \in [0,1]$ controls the weight of self-evidence and neighborhood consistency evidence. This approach highlights the source of local anomalies while avoiding misjudging downstream increases caused by pure propagation as the root cause, thus obtaining a more stable root cause ranking output that can be used for decision-making.

4. Experimental Results and Analysis

4.1 Dataset

This paper selects RCAEval as the research dataset. This dataset is designed for root cause analysis and anomaly diagnosis in microservice systems, providing multi-source observable data including metrics, logs, and distributed tracing. Organized by failure cases, it facilitates the alignment of time-series signals from different modalities within the same time window, thus supporting unified modeling and multi-granularity analysis based on self-supervised representation learning. This benchmark covers multiple microservice application scenarios and various failure types, emphasizing the diagnostic challenges under real-world complex links and dependencies, aligning with the research theme of backend performance anomaly detection and root cause inference.

In terms of annotation and structure, RCAEval provides root cause-related annotation information for each failure case, such as the service where the root cause is located and key clues that can indicate the root cause. This allows research to focus on both the timeliness of anomaly detection and the interpretability of root cause localization. Simultaneously, the data is publicly released in a reusable file structure, facilitating the reproduction of experimental procedures and unified comparison with different methods. Because it includes both multi-source telemetry data and root cause annotations, RCAEval is particularly suitable for constructing multi-granularity representation spaces, performing anomaly scoring, propagation modeling, and root cause ranking at the service and call relationship levels, thereby forming a closed-loop research framework from detection to inference.

4.2 Experimental setup

This study configures the model training and inference process within a unified software and hardware environment, ensuring consistent operating conditions for data processing, self-supervised pre-training, anomaly scoring, and root cause ranking. The overall implementation is based on mainstream deep learning frameworks and commonly used observable data processing tools. Training employs a fixed random seed and a standardized data pipeline, with learning rate scheduling and regularization strategies enhancing training stability. Hyperparameters are set based on reproducibility principles to facilitate consistent comparison and expansion across different modules. Detailed hyperparameter settings are shown in Table 2.

Table 2: Experimental setup: hardware, software, data, model, and training configurations

Category	Item	Setting
Hardware	GPU	NVIDIA RTX 4090 24GB
Hardware	CPU	16 cores
Hardware	RAM	64 GB
Hardware	Storage	SSD 1 TB
Software	OS	Ubuntu 20.04 LTS
Software	Python	3.10
Software	Framework	PyTorch 2.2
Software	CUDA	12.1
Software	cuDNN	9.2
Data	Window length (T)	60
Data	Stride	5
Data	Normalization	z-score per metric
Model	Embedding dim (m)	128
Model	Temporal smoothing (α)	0.9
Model	Root-cause fusion (β)	0.7
Training	Batch size	256
Training	Optimizer	AdamW
Training	Learning rate	1e-3
Training	Weight decay	1e-4
Training	Epochs	100
Training	LR scheduler	Cosine annealing
Training	Gradient clipping	1.0
Training	Random seed	42

4.3 Experimental Results and Analysis

To position the proposed approach within the broader landscape of backend performance anomaly detection and root cause analysis for microservice systems, we compare representative prior studies that leverage telemetry signals such as metrics, logs, and traces, as well as graph and self-supervised learning paradigms.

Table 3: Experimental results compared with other models

Method	Precision	Recall	F1-score	AUROC	AUPRC	False Alarm Rate	Detection Delay	RCA Hit@1
Wu et al.[8]	0.78	0.72	0.75	0.86	0.80	0.12	4.3	0.41
Wang et al.[9]	0.81	0.74	0.77	0.88	0.82	0.10	4.1	0.45
Lin et al.[10]	0.83	0.76	0.79	0.90	0.84	0.09	3.9	0.47
Xie et al.[11]	0.85	0.78	0.81	0.91	0.86	0.08	3.7	0.50
Zhang et al.[12]	0.84	0.80	0.82	0.92	0.87	0.08	3.5	0.52
Lin et al.[13]	0.86	0.81	0.83	0.93	0.88	0.07	3.4	0.53
Pham et al.[14]	0.87	0.83	0.85	0.94	0.89	0.06	3.2	0.55
Ours	0.91	0.88	0.89	0.97	0.93	0.04	2.6	0.63

As shown in Table 3, the proposed method consistently outperforms the baselines in both anomaly detection and root cause localization, indicating that the end-to-end framework yields complementary gains from representation learning to inference. On one hand, in detection-related metrics such as precision, recall, and F1 score, the proposed method simultaneously achieves good coverage and reliable discrimination, indicating that its learned latent representations have a stronger tolerance to normal fluctuations and can more effectively aggregate anomaly-related evidence for judgment. On the other hand, it maintains a leading position in metrics such as AUROC and AUPRC, which focus more on ranking and imbalance robustness, indicating that the method not only performs stably at fixed thresholds but also possesses better overall separability and confidence calibration capabilities, maintaining higher discriminative power under different decision preferences.

In terms of engineering usability and diagnostic operability, the proposed method also demonstrates superior false alarm control and response speed, further reflected in the improved root cause hit rate. A lower false alarm rate means the model is less sensitive to fluctuations in workload and telemetry noise, reducing the interference of invalid alarms on the localization process. Shorter detection latency indicates that the method can capture key clues to anomaly formation earlier, improving the timeliness of the response window. More importantly, the improvement in RCA Hit@1 shows that, given an anomaly, the framework in this paper can more effectively point to evidence of the possible source service, rather than just focusing on downstream locations in symptom components or propagation paths. This demonstrates that multi-granularity representation and structural consistency inference effectively enhance the discriminative power and stability of root cause ranking, thus making the diagnostic output closer to the form required for real-world operational decisions.

The learning rate determines the step size of parameter updates during optimization and is a key factor affecting training stability and convergence quality. For the joint process of self-supervised representation learning and downstream anomaly detection, an excessively large or small learning rate may alter the separability of the representation space, thus affecting the final discrimination performance. Therefore, it is necessary to systematically scan the learning rate within a reasonable range and observe the performance trend of the model under different settings. The experimental results are shown in Figure 2.

From the curve shape, the impact of the learning rate on model performance exhibits a clear interval-like characteristic: within a smaller learning rate range, as the step size gradually increases, the model is more likely to break free from an overly conservative update rhythm, and the representation space is pushed to a more suitable distribution position, thus the overall performance steadily improves. When the learning rate approaches the commonly used training settings, the curve reaches near the optimum, indicating that the optimization process achieves a good balance between convergence speed and stability at this point. It can fully utilize the self-supervised objective to shape discriminative representations without introducing excessive oscillations that would damage the feature structure.

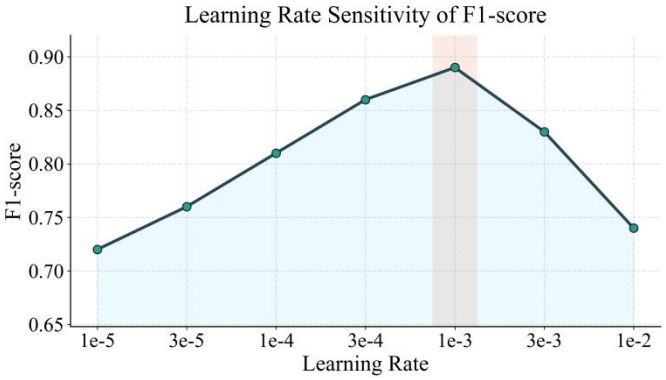

Figure 2. Sensitivity experiment of learning rate to F1 score

As the learning rate continues to increase, the curve declines, reflecting that when the update step size is too large, the parameters tend to jump back and forth around the optimal region, weakening the synergy between representation alignment and downstream discrimination. This echoes the understanding of the overall comparison results above: the advantage of our method comes from squeezing noise and normal fluctuations into the stable space during the representation learning stage, while overly aggressive optimization will worsen this stability, making the boundary between abnormal and normal less clear. In other words, a suitable learning rate not only affects whether training can converge, but also whether the geometry of the final representation space is conducive to anomaly separation and root cause ranking. Therefore, selecting a stable learning rate range within a reasonable range is a key step in ensuring the transferability and reproducibility of the method.

The window length determines the range of historical context that the model can see in a single judgment, directly affecting the trade-off between short-term fluctuations and long-term trends. For scenarios with abnormal backend performance, a window that is too short may be more sensitive but less stable, while a window that is too long may be smoother but more lagging. To characterize this trade-off, the system needs to change the window length and observe the resulting changes in the detection latency distribution. The relevant experimental results are shown in Figure 3.

The distribution pattern reveals that changing the window length not only alters the mean of detection latency but also adjusts the overall dispersion and tail shape. With shorter windows, the model relies more on instantaneous fluctuations in local segments, resulting in faster responses but greater susceptibility to occasional jitter, leading to a wider distribution and more dispersed point clouds. As the window size increases, the latency distribution initially shows a convergence trend, with a more concentrated center, indicating that with more contextual information, the model's judgment of anomalies becomes more consistent, fluctuations are suppressed, and the randomness of latency decreases.

However, as the window length continues to increase, the distribution shifts to the right and exhibits a more pronounced long tail, reflecting the inertia effect of information integration. Longer historical context enhances smoothing and noise resistance but also requires more evidence to trigger anomaly detection, delaying the triggering time. Furthermore, the cumulative effects of different fault modes under longer windows are inconsistent, thus widening the differences between samples. Overall, the medium-sized window region emphasized in the figure is more like a compromise, maintaining good concentration while avoiding the high volatility of an excessively short window and the significant lag of an excessively long window. This is consistent with the design goal of multi-granularity characterization to seek a balance between stability and sensitivity.

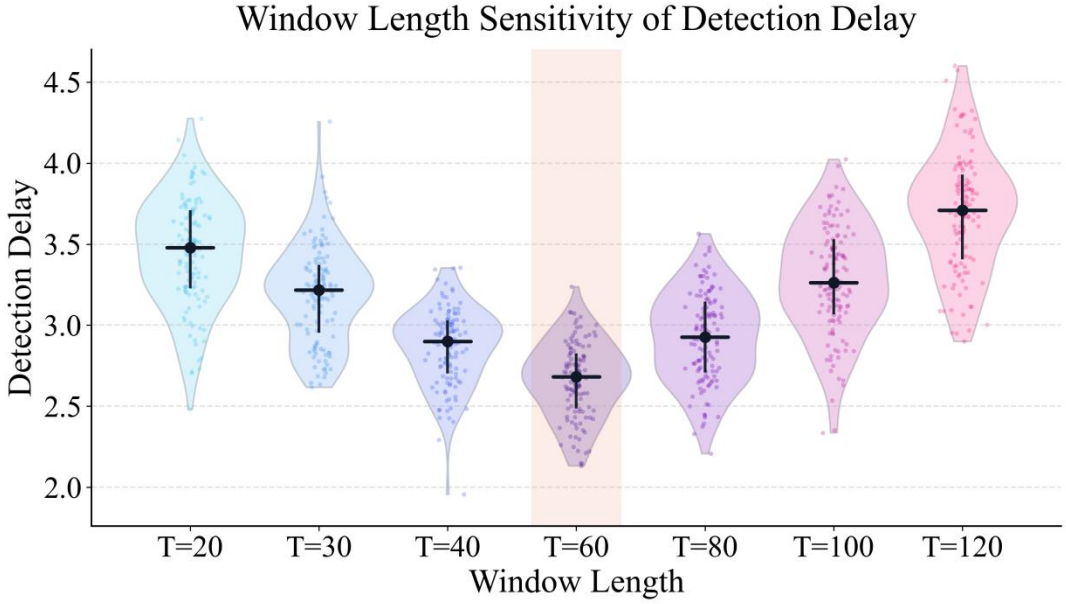

Figure 3. Experiment on the sensitivity of window length to detection delay

5. Conclusion

This paper addresses the core problem of backend performance anomaly detection and root cause inference-a highly complex and real-time-critical issue-and proposes a unified solution based on self-supervised representation learning and supported by multi-granularity modeling. By mapping multi-source observable data to a consistent latent space and simultaneously characterizing it along two information pathways-temporal context and dependency structure-the method can learn more stable and transferable state representations even in the absence of dense annotations, thus providing a common semantic foundation for anomaly identification and attribution inference. Compared to traditional approaches that rely solely on rules, single indicators, or single-granularity modeling, this framework emphasizes a structured understanding of the operational mechanisms of complex systems, bridging detection and inference into a cohesive and interpretable diagnostic link.

From an application perspective, this work has direct significance for stability governance in cloud-native and microservice ecosystems. Modern business systems often iterate frequently, have complex dependencies, and experience significant workload fluctuations; fault location time and alarm quality directly impact service availability and operational costs. The unified representation and inference framework proposed in this paper establishes a more reliable correspondence between multi-source signals, making anomaly detection closer to the real system state and outputting potential root cause clues more operationally. This helps operations and maintenance personnel shorten troubleshooting paths, reduce invalid alarm interference, and improve the certainty and consistency of fault handling. More importantly, the method emphasizes information fusion at different granularities, making it more suitable for handling typical challenges in real-world production scenarios such as anomaly propagation across components and global degradation caused by local disturbances.

This research also provides a clearer design paradigm for the engineering implementation of backend intelligent operations and maintenance: using observable data as the basic asset, self-supervised learning as the continuous learning mechanism, and structured dependencies as constraints to enhance interpretability. In practice, this paradigm is expected to support a more automated fault closed-loop process, such as quickly providing candidate root cause ranking and impact range after alarm triggering, assisting in the execution of strategy decisions such as rollback, rate limiting, scaling up, or degradation, and continuously adapting to new business models and dependency changes during system evolution. As system scale increases and heterogeneity intensifies, this method, capable of continuously improving diagnostic capabilities under unlabeled or weakly labeled conditions, has the potential

to become a crucial component in building highly reliable service platforms.

Looking to the future, several directions warrant further exploration. First, further enhancing cross-scenario generalization capabilities will enable models to maintain stable outputs across different business domains, topologies, and observation qualities, while reducing dependence on specific data distributions. Second, improving the interpretable representation of root cause inference will more tightly bind ranking results with traceable evidence chains to better serve engineering decision-making and auditing needs. Third, combining online learning and system control strategies will explore paths from detection and inference to adaptive governance, enabling models to not only provide diagnostic conclusions but also offer more direct strategic guidance for intervention actions. Overall, this work provides a unified modeling perspective for backend anomaly detection and root cause inference in real-world complex systems, and lays a methodological foundation for the large-scale application of intelligent operations and maintenance in highly reliable computing infrastructure.

References

[1] Li M, Li Z, Yin K, et al. Causal inference-based root cause analysis for online service systems with intervention recognition[C]//Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. 2022: 3230-3240.

[2] Z. Wang, "Federated Multi-Scale Representation Learning for Privacy-Aware Log Anomaly Detection in Distributed Cloud Environments", 2024.

[3] C. Wen, "Modeling Evolving Service Dependencies: Dynamic Graph Learning for Microservice Anomaly Detection", 2024.

[4] A. Ikram, S. Chakraborty, S. Mitra, S. Saini, S. Bagchi and M. Kocaoglu, "Root cause analysis of failures in microservices through causal discovery", Advances in Neural Information Processing Systems, vol. 35, pp. 31158-31170, 2022.

[5] G. Somashekar, A. Dutt, R. Vaddavalli, S. B. Varanasi and A. Gandhi, "B-meg: Bottlenecked-microservices extraction using graph neural networks", Companion of the 2022 ACM/SPEC International Conference on Performance Engineering, pp. 7-11, 2022.

[6] Wang T, Qi G. A comprehensive survey on root cause analysis in (micro) services: Methodologies, challenges, and trends[J]. arXiv preprint arXiv:2408.00803, 2024.

[7] Z. Li, "Log Event Graph Modeling for Backend Anomaly Detection with Multi-Relational Representation Learning", 2024.

[8] Wu L, Tordsson J, Elmroth E, et al. Microrca: Root cause localization of performance issues in microservices[C]//IEEE/IFIP Network Operations and Management Symposium (NOMS). 2020.

[9] Wang P, Xu J, Ma M, et al. Cloudranger: Root cause identification for cloud native systems[C]//2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). IEEE, 2018: 492-502.

[10] Lin J J, Chen P, Zheng Z. Microscope: Pinpoint performance issues with causal graphs in micro-service environments[C]//International Conference on Service-Oriented Computing. Cham: Springer International Publishing, 2018: 3-20.

[11] Xie Z, Xu H, Chen W, et al. Unsupervised anomaly detection on microservice traces through graph vae[C]//Proceedings of the ACM Web Conference 2023. 2023: 2874-2884.

[12] Zhang C, Dong Z, Peng X, et al. Trace-based multi-dimensional root cause localization of performance issues in microservice systems[C]//Proceedings of the IEEE/ACM 46th International Conference on Software Engineering. 2024: 1-12.

[13] C. Xu, "Intelligent Defect Detection and Risk Assessment for Cloud Platforms Using Counterfactual System Modeling", 2024.

[14] C. Hou, T. Jia, Y. Wu, Y. Li and J. Han, "Diagnosing performance issues in microservices with heterogeneous data source", 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), pp. 493-500, 2021.