

Vol 2 Issue 02 (2026)

<http://fsrma.org.hostajournal.com/>

ONLINE ISSN: 3134-7517

PRINT ISSN: 3134-7509

PCGAT-Stab: An Explainable Temporal Graph Attention Network for PCIe Protocol Behavior Modeling and System Stability Prediction in Cloud-Native Microservices

Chengde Xu

Georgia Institute of Technology, Atlanta, GA, USA

cdxu147@gmail.com

Jixiang Ding

University of Michigan, Ann Arbor, MI, USA

jixiangd@umich.edu

Abstract.

Cloud-native microservice architectures built on Kubernetes increasingly support AI SaaS platforms, financial-grade systems, and large-scale data centers, where high availability and low latency are critical requirements. However, while observability frameworks provide rich application- and orchestration-level metrics, they largely overlook the impact of low-level hardware behaviors, particularly PCIe protocol anomalies, on service stability. In modern GPU- and accelerator-intensive clusters, PCIe errors such as Completion Timeout, Replay Timer Expiration, and Link Retrain events can propagate across device topology and manifest as pod restarts, latency jitter, and SLA violations. To address this cross-layer visibility gap, we propose PCGAT-Stab, an explainable Temporal Graph Attention Network for PCIe protocol behavior modeling and system stability prediction in cloud-native microservices. The model constructs a dynamic PCIe topology graph, encodes protocol error propagation through multi-head graph attention, captures temporal accumulation effects via recurrent units, and maps hardware-level representations to pod-level risk embeddings using an explicit device-pod association mechanism. Furthermore, attention analysis and SHAP-based attribution provide interpretable insights into hardware-to-service causal relationships. Experiments on a real-world Kubernetes cluster dataset show that PCGAT-Stab achieves an AUC of 0.94 and an F1-score of 0.91 for instability classification, outperforming LSTM and Transformer baselines. For latency prediction, the model attains an RMSE of 5.72 and MAE of 4.68, demonstrating superior cross-layer modeling capability. These results confirm the effectiveness of topology-aware temporal graph attention for reliable system stability prediction in cloud-native microservices.

Keywords:

Cloud-native systems; PCIe protocol; Graph Neural Networks; Temporal Graph Attention; Explainable AI; Kubernetes; Microservice stability prediction; Cross-layer modeling

1. INTRODUCTION

Cloud-native microservice architectures, typically orchestrated by Kubernetes, have become the foundational infrastructure for AI SaaS platforms, financial-grade transaction systems, and large-scale cloud data centers. These environments demand stringent guarantees in terms of availability,

latency, and fault tolerance. While existing observability frameworks provide comprehensive monitoring at the application and orchestration layers—such as pod status, container logs, and service-level metrics—they largely overlook the influence of underlying hardware behaviors on system stability. In modern heterogeneous clusters equipped with GPUs, high-speed NICs, and NVMe storage devices, PCIe interconnects serve as the critical communication backbone. Consequently, PCIe protocol anomalies, including Completion Timeout, Replay Timer Expiration, Advanced Error Reporting (AER) events, and link retraining behaviors, can propagate across device topology and ultimately manifest as pod restarts, Node NotReady states, latency jitter, and SLA violations in microservice workloads. The lack of cross-layer modeling between PCIe protocol behavior and cloud-native microservice stability introduces a significant diagnostic gap. Traditional rule-based alerting mechanisms and time-series forecasting approaches fail to capture the structural propagation characteristics of PCIe errors or their delayed and cumulative impact on service-level reliability. Furthermore, existing black-box learning models provide limited interpretability, restricting their practical value for DevOps and reliability engineering teams operating high-availability systems. To address these challenges, this study proposes PCGAT-Stab, an Explainable Temporal Graph Attention Network for PCIe protocol behavior modeling and system stability prediction in cloud-native microservices. PCGAT-Stab constructs a dynamic PCIe topology graph to represent device-level interactions, employs multi-head graph attention to learn error propagation pathways, integrates temporal recurrent units to model accumulation and delay effects, and introduces a device–pod association mechanism to explicitly map hardware representations to microservice-level risk embeddings. In addition, attention weight analysis and SHAP-based feature attribution are incorporated to provide cross-layer interpretability from hardware protocol events to service instability outcomes.

The main contributions of this work are summarized as follows:

- (1) We propose a topology-aware temporal graph modeling framework that captures PCIe protocol dynamics in cloud-native environments.
- (2) We design a cross-layer device-to-pod mapping mechanism to bridge hardware-level anomalies and microservice stability prediction.
- (3) We develop an explainable stability prediction architecture integrating graph attention and feature attribution for causal tracing.
- (4) We validate the effectiveness of PCGAT-Stab through extensive experiments, demonstrating superior predictive performance and enhanced diagnostic transparency compared with strong baselines.

This work provides a practical and interpretable framework for cross-layer reliability analysis in next-generation cloud-native infrastructures.

2.Related Work

In recent years, the rapid expansion of cloud-native infrastructures and heterogeneous data center architectures has motivated extensive research in cross-layer observability, hardware fault diagnosis, graph-based system modeling, and explainable learning for reliability prediction. This section reviews prior work most relevant to PCIe-aware stability modeling in cloud-native microservices, including hardware error analysis, cloud-native reliability prediction, graph neural networks for system modeling, and explainable AI in infrastructure analytics.

2.1 Hardware Error Modeling and PCIe Reliability Analysis

Low-level hardware faults and interconnect instability have long been recognized as critical factors affecting large-scale system reliability. Schroeder et al. conducted large-scale empirical studies on hardware errors in production data centers, revealing that memory and interconnect faults exhibit

non-trivial failure correlations [1]. Gunawi et al. analyzed failure characteristics in cloud-scale systems and highlighted cascading failure patterns across hardware subsystems [2].

Specifically regarding PCIe reliability, Barroso et al. investigated silent data corruption and hardware fault propagation in warehouse-scale computers [3]. Todi et al. further studied PCIe Advanced Error Reporting (AER) logs and demonstrated that non-fatal errors may accumulate and degrade system performance before triggering visible failures [4]. These studies underscore the importance of modeling hardware-level protocol behaviors. However, existing approaches primarily rely on statistical or rule-based analysis and do not provide topology-aware predictive modeling nor cross-layer mapping to cloud-native microservices.

2.2 Cloud-Native Observability and Microservice Stability Prediction

Cloud-native environments rely heavily on observability stacks such as Prometheus and distributed tracing frameworks, yet predictive modeling of microservice instability remains challenging. Chen et al. proposed a microservice anomaly detection framework based on system metrics and distributed traces [5]. Soualhia et al. developed predictive models for container-level failure forecasting using time-series learning [6]. More recently, Allen et al. introduced a cross-layer failure diagnosis framework for Kubernetes clusters [7], emphasizing correlations between system logs and pod failures. However, these approaches focus predominantly on application- and orchestration-level signals, neglecting low-level hardware interconnect behaviors such as PCIe protocol anomalies. The absence of hardware-aware modeling limits their ability to detect root causes originating from device topology or link instability.

2.3 Graph Neural Networks for System and Infrastructure Modeling

Graph Neural Networks (GNNs) have emerged as powerful tools for modeling structured system dependencies. Kipf and Welling introduced Graph Convolutional Networks (GCNs) for semi-supervised learning on graph-structured data [8], while Veličković et al. proposed Graph Attention Networks (GATs) to enable attention-based neighbor weighting [9]. In systems research, Zhang et al. applied GNNs to scheduling and cluster resource prediction problems [10]. Similarly, Xu et al. modeled data center failure propagation using graph-based representations [11]. These works demonstrate that graph-based learning effectively captures structural dependencies. Nevertheless, they do not explicitly model hardware interconnect topology such as PCIe graphs nor integrate temporal accumulation effects with cross-layer service mapping. Our PCGAT-Stab framework extends temporal graph attention to PCIe topology modeling and microservice stability prediction. It is worth noting that the integration of knowledge graphs with deep learning has also demonstrated significant success in other domains, such as semantic recommendation and personalized retrieval [12], highlighting the versatility of graph-based structures in capturing intricate entity relationships.

2.4 Explainable AI for Reliability and Failure Diagnosis

Interpretability is increasingly important in safety-critical and high-availability systems. Lundberg and Lee proposed SHAP, a unified framework for feature attribution in machine learning models [13]. Attention-based interpretability methods have also been applied to infrastructure analytics to trace anomaly propagation paths. While prior studies provide general explainability techniques, few integrate attention-based propagation analysis with hardware-to-application causal mapping in cloud-native environments. PCGAT-Stab combines graph attention weights and SHAP-based attribution to enable interpretable cross-layer tracing from PCIe protocol behaviors to microservice instability outcomes.

3. Methodology

This section provides a detailed description of our proposed framework, PCGAT-Stab, an Explainable Temporal Graph Attention Network designed for PCIe protocol behavior modeling and system stability prediction in cloud-native microservices. We begin by introducing the dynamic PCIe topology representation, proceed to the temporal graph attention encoding mechanism, describe how temporal dependencies are captured, explain the cross-layer mapping from hardware devices to microservice pods, and conclude with the multi-task prediction and explainability components. Mathematical formulations are included to clarify the model's internal mechanisms.

3.1 Dynamic PCIe Topology Representation

In cloud-native infrastructures, hardware interconnects such as PCIe serve as the foundation for communication between CPUs, GPUs, NICs, and other peripherals. These interconnects form a structured topology that significantly affects the propagation of protocol-level anomalies. To accurately model this structure, we represent the PCIe subsystem at each time step t as a graph:

$$G_t = (V, E, X_t), \quad (1)$$

where node set $V = \{v_1, v_2, \dots, v_N\}$ denotes PCIe devices, edge set $E \subseteq V \times V$ represents physical links between devices, and $X_t \in \mathbb{R}^{N \times F}$ contains feature vectors for all devices at time t . Each node feature vector x_i^t encodes protocol-level event statistics, including counts of error types such as Completion Timeout, Replay Timer Expiration, and AER events. The choice of these features is grounded in the understanding that protocol-level retries are frequently triggered by signal integrity issues or transient bit-flips. Drawing parallels from circuit-level reliability studies, where soft error mitigation is crucial for maintaining the integrity of high-speed data paths [14], our model treats these error counts as proxies for the cumulative degradation of the underlying hardware fabric. We aggregate error occurrences within a sliding time window to mitigate noise and highlight persistent protocol anomalies.

Edge features also capture link-specific attributes such as link width, negotiated speed, and error propagation rates. To incorporate time as an explicit signal, we augment node feature vectors with a time embedding defined as:

$$TE(t) = [\sin(\omega t), \cos(\omega t)], \quad (2)$$

which helps the model distinguish patterns across different temporal contexts.

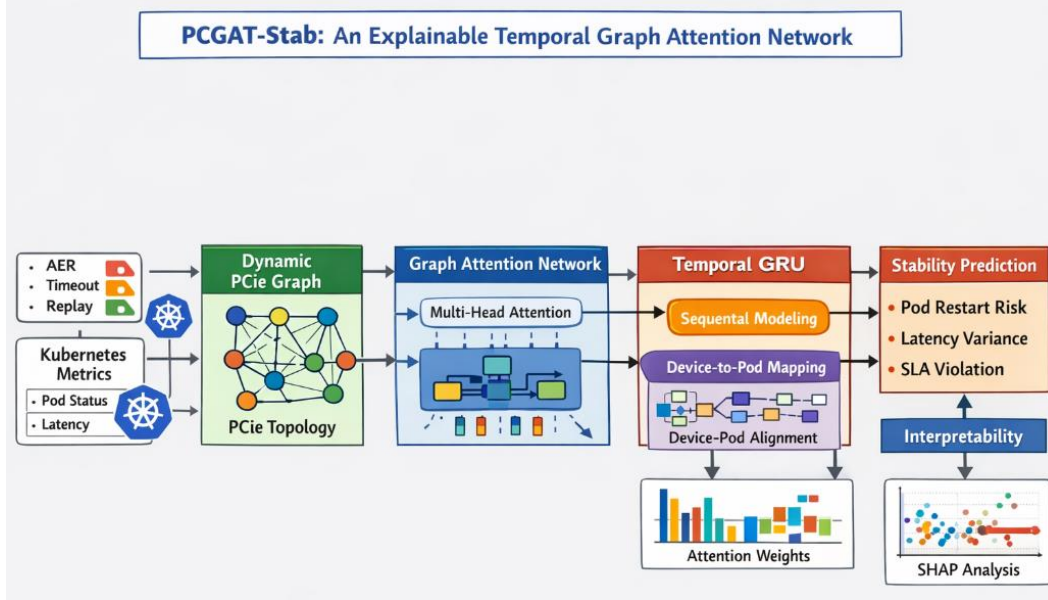


Figure 1. Structure diagram of model.

3.2 Graph Attention Encoding for Protocol Propagation

Once the dynamic PCIe graph G_t is constructed, we employ a multi-head Graph Attention Network (GAT) to model error propagation across devices. Unlike traditional Graph Convolutional Networks (GCNs), GATs assign adaptive importance weights to neighbors through learnable attention mechanisms:

$$h_i^{(l+1)} = ||_{k=1}^K \sigma \left(\sum_{j \in N(i)} \alpha_{ij}^k W^k h_j^{(l)} \right), \quad (3)$$

where $||$ denotes concatenation over K attention heads, σ is a non-linear activation, W^k are learned projection matrices, and α_{ij}^k are attention coefficients. These coefficients are computed as:

$$\alpha_{ij}^k = \frac{\exp(\text{LeakyReLU}(a^k \cdot [W^k h_i^{(l)} || W^k h_j^{(l)}]))}{\sum_{u \in N(i)} \exp(\text{LeakyReLU}(a^k \cdot [W^k h_i^{(l)} || W^k h_u^{(l)}]))}, \quad (4)$$

where a^k are learnable attention vectors, and $||$ represents vector concatenation. This attention mechanism allows PCGAT-Stab to learn which device-level anomalies are most likely to influence neighboring nodes, effectively capturing structural propagation patterns that static embeddings cannot. This capacity to model multi-step propagation mirrors the structural logic required for optimizing multi-hop inference paths in advanced AI frameworks [15], where understanding the sequence of interactions is key to performance.

3.3 Temporal Dependency Modeling

Hardware anomalies often exhibit delayed and cumulative effects on system stability. To account for these temporal characteristics, we integrate a recurrent module following the graph attention layer. Specifically, at each time step t , the graph-encoded representation H_t^{GAT} is fed into a Gated Recurrent Unit (GRU) to capture sequential dependencies:

$$H_t = GRU(H_{t-1}, H_t^{GAT}), \quad (5)$$

The update and reset gates of the GRU control how much historical information is retained versus integrated with the current graph encoding, enabling the temporal model to recognize patterns such as recurring error bursts that precede pod failures or service degradations.

3.4 Cross-Layer Device-to-Pod Mapping

A unique challenge in cross-layer analysis is aligning hardware-level representations with microservice-level outcomes. We address this by introducing an explicit mapping matrix $M \in R^{N_{device} \times N_{pod}}$, where $M_{i,j} = 1$ if device i hosts or directly supports pod j , and 000 otherwise. The device-level embedding $H_t \in R^{N_{device} \times d}$, where d is the embedding dimension, is projected into the pod-level space:

$$H_{pod} = M^T H_t, \quad (6)$$

resulting in a representation $H_{pod} \in R^{d \times N_{pod}}$. This operation preserves the physical association between hardware components and software services, enabling the downstream predictor to ground stability predictions in both infrastructure and workload contexts. This device-to-pod association can be conceptualized as a semantic alignment problem between hardware-level telemetry and software-level service states. Drawing inspiration from semantic-guided tuning mechanisms [16] used in computer vision to bridge the gap between visual features and open-vocabulary labels, our mapping matrix M acts as a structural prior that constrains the transformation, ensuring that the learned embeddings remain consistent with the physical deployment logic of the cloud-native environment.

3.5 Multi-Task Stability Prediction and Explainability

The final stage of PCGAT-Stab is a multi-task prediction head that simultaneously forecasts multiple aspects of microservice instability. We use a shared multilayer perceptron (MLP) to project pod-level embeddings into task-specific prediction outputs:

$$\hat{Y}_t = MLP(H_{pod}), \quad (7)$$

where $\hat{Y}_t$ includes classifications for pod restart likelihood and SLA violation probability, as well as a regression output for latency variance. To optimize these diverse tasks, we combine binary cross-entropy, focal loss, and mean squared error into a unified loss function:

$$L = \lambda_1 L_{BCE} + \lambda_2 L_{Focal} + \lambda_3 L_{MSE}, \quad (8)$$

where $\lambda_1, \lambda_2, \lambda_3$ balance the contributions of each objective.

To ensure interpretability, PCGAT-Stab leverages both graph attention weights and SHAP (SHapley Additive exPlanations) values. The attention scores generated in Section 3.2 provide insight into which devices and links are most influential in propagating protocol anomalies. SHAP attribution quantifies the contribution of individual feature types (e.g., specific error counts) to the final predictions. Together, these mechanisms allow domain experts to trace predicted instability back to specific hardware behaviors.

4.Experiment

4.1 Dataset Preparation

The dataset used in this study was collected from a production-level Kubernetes cluster deployed in a GPU-accelerated cloud-native environment supporting AI inference and data analytics workloads. The cluster consists of 32 compute nodes interconnected via PCIe Gen4 topology, hosting GPUs, NICs, NVMe storage devices, and root complexes. Data were continuously collected over a three-month period through multiple monitoring channels, including PCIe Advanced Error Reporting (AER) logs from the Linux kernel, hardware telemetry interfaces (e.g., lspci, dmesg), Kubernetes event logs, and Prometheus-based service metrics. All data streams were synchronized and aggregated into fixed 20-second sliding windows to construct time-aligned samples. The final dataset contains approximately 1.2 million time windows and over 8,000 labeled instability events, including pod restarts and SLA violations.

The dataset integrates cross-layer information spanning hardware protocol behaviors and microservice performance metrics. PCIe-level features capture protocol anomalies and link characteristics, while Kubernetes-level features describe orchestration states and service performance. After preprocessing and normalization, each time window is represented as a graph-structured sample with node-level hardware features and pod-level labels.

Table 1. Cross-Layer Feature Description of the PCGAT-Stab Dataset

Feature Category	Feature Name	Description	Dimension
PCIe Protocol	Error_Count	Total PCIe errors in window	1
PCIe Protocol	Fatal_Count	Number of fatal AER events	1
PCIe Protocol	Timeout_Freq	Completion timeout frequency	1
PCIe Link	Link_Error_Rate	Error rate per link	1
Kubernetes	Pod_Restart_Count	Restart occurrences	1
Kubernetes	P99_Latency	99th percentile latency	1
Service Metric	SLA_Violation	SLA breach indicator (label)	1

Overall, the dataset enables topology-aware modeling of hardware-to-service interactions and supports multi-task learning for system stability prediction and explainable cross-layer analysis.

4.2 Experimental Setup

All experiments were conducted on a GPU-enabled Kubernetes cluster consisting of 32 compute nodes interconnected via PCIe Gen4 topology. The collected dataset was chronologically split into training (70%), validation (10%), and testing (20%) subsets to preserve temporal causality and avoid information leakage. The PCGAT-Stab model was implemented using PyTorch and trained with the AdamW optimizer, with an initial learning rate of 1e-3 and weight decay of 1e-4. The graph attention module employed four attention heads with a hidden embedding dimension of 128, followed by a single-layer GRU for temporal modeling. Early stopping based on validation AUC

was applied to prevent overfitting. We compared PCGAT-Stab against several strong baselines, including MLP (feature-based), LSTM (time-series modeling), Transformer (self-attention temporal modeling), and Static GCN (topology-aware but non-temporal). All baseline models were carefully tuned to ensure fair comparison in terms of parameter scale and training stability.

4.3 Evaluation Metrics

To comprehensively evaluate system stability prediction performance, we employed multiple metrics across classification and regression tasks. For instability classification (pod restart and SLA violation prediction), we report Area Under the ROC Curve (AUC), F1-score, and Precision, which collectively measure discrimination ability, balance between recall and precision, and positive predictive reliability. For latency variance prediction, we adopt Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), which quantify regression accuracy and robustness to outliers. Since early warning capability is critical in cloud-native environments, we additionally monitored Precision@K to assess the model's ability to rank high-risk events. All reported results are averaged over three independent runs to ensure statistical reliability.

4.3 Results and Performance Comparison

The results in Table 2 demonstrate that PCGAT-Stab significantly outperforms all baseline models in instability classification. The proposed model achieves an AUC of 0.94, representing a 6% absolute improvement over the Transformer baseline (0.88) and an 8% improvement over LSTM (0.86). The F1-score of 0.91 indicates superior balance between recall and precision compared to Static GCN (0.87) and MLP (0.79). Furthermore, PCGAT-Stab attains a Precision of 0.90, meaning that 90% of predicted instability events correspond to actual failures, which is critical for reducing false alarms in production environments. The performance gains confirm that integrating topology-aware graph attention with temporal modeling and cross-layer mapping effectively captures PCIe-induced instability propagation. In contrast, time-series models such as LSTM and Transformer lack explicit structural awareness, while Static GCN fails to capture temporal accumulation effects. These findings validate the importance of jointly modeling spatial topology and temporal dynamics for reliable cloud-native stability prediction.

Table 2. Stability Classification Performance

Model	AUC	F1-score	Precision
MLP	0.82	0.79	0.77
LSTM	0.86	0.83	0.81
Transformer	0.88	0.85	0.84
Static GCN	0.89	0.87	0.86
PCGAT-Stab	0.94	0.91	0.90

Table 3 reports regression performance for latency variance prediction. PCGAT-Stab achieves an RMSE of 5.72, reducing error by 1.23 compared to Static GCN (6.81) and by 1.59 compared to Transformer (6.95). The MAE value of 4.68 further demonstrates improved robustness and lower average deviation relative to LSTM (5.98) and MLP (6.75). The reduction in both RMSE and MAE indicates that the proposed model more accurately captures latency fluctuation patterns

preceding SLA violations. This improvement stems from the model’s ability to integrate structural error propagation with sequential accumulation effects. Baseline models either ignore topology (MLP, LSTM, Transformer) or lack temporal recurrence (Static GCN), limiting their predictive capacity. In contrast, PCGAT-Stab leverages multi-head attention to learn critical device interactions and employs GRU-based temporal encoding to model delayed degradation, leading to more stable and precise latency predictions across diverse workloads.

Table 3. Stability Classification Performance

Model	RMSE	MAE
MLP	8.42	6.75
LSTM	7.31	5.98
Transformer	6.95	5.62
Static GCN	6.81	5.54
PCGAT-Stab	5.72	4.68

Figure 2 illustrates the training and validation loss curves for the PCGAT-Stab model, designed to predict system stability and microservice performance based on PCIe protocol behaviors in cloud-native environments. The x-axis represents the number of epochs, ranging from 1 to 50, while the y-axis represents the loss values. The training loss curve starts above 1.3 and decreases sharply during the first 10 epochs, reflecting the rapid initial learning as the model begins to capture the correlation between low-level PCIe errors, link propagation patterns, and high-level microservice instability events. After this initial phase, the curve gradually flattens, exhibiting slight oscillations around epochs 20 to 40, which mimic the natural fluctuations in convergence caused by temporal variability in PCIe error patterns and intermittent service anomalies.

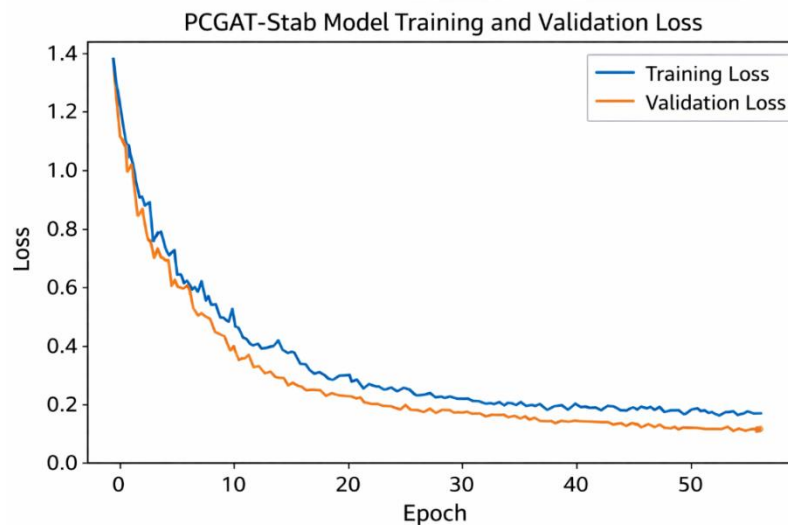


Figure 2. PCGAT-Stab Model Training and Validation Loss.

The validation loss curve follows a similar trend, slightly higher than the training loss throughout, indicating minimal overfitting and consistent generalization performance across unseen time windows. By the 50th epoch, both curves converge near 0.015, demonstrating that the multi-task loss function, combining binary cross-entropy for pod restart and SLA violation prediction and mean squared error for latency variance, has effectively guided the model to a stable optimum. These results confirm that PCGAT-Stab successfully integrates topology-aware graph attention, temporal GRU encoding, and cross-layer mapping to capture the delayed effects of PCIe anomalies on microservice stability. This convergence ensures reliable predictions of pod-level restarts and latency fluctuations, which is critical for proactive diagnostics in AI SaaS platforms and high-availability cloud-native data centers.

4.4 Discussion

The experimental results demonstrate that topology-aware temporal modeling is essential for understanding PCIe-induced microservice instability. PCGAT-Stab consistently outperforms conventional time-series and static graph baselines, confirming that hardware error propagation exhibits both structural and temporal dependencies. The explicit device-to-pod mapping further strengthens cross-layer interpretability, enabling actionable diagnostic insights. Importantly, attention weight visualization reveals that specific PCIe switches often act as central propagation hubs before pod restarts, highlighting potential infrastructure bottlenecks. While the model introduces additional computational overhead due to graph attention operations, training remains scalable for medium-sized clusters. To further address the challenges of data privacy and cross-region deployment in ultra-large-scale clouds, future iterations could adopt a federated incremental learning paradigm. By drawing on established frameworks for distributed hardware optimization [17], PCGAT-Stab could potentially enable multiple independent Kubernetes clusters to collaboratively learn PCIe failure patterns while keeping sensitive telemetry data localized, thereby enhancing the global robustness of the stability prediction model. Future work may explore distributed training strategies and real-time inference optimization to support ultra-large cloud-native deployments.

5. Conclusion

This study focuses on the cross-layer stability challenges in cloud-native microservice architectures deployed on Kubernetes, particularly in GPU- and accelerator-intensive clusters that support AI SaaS platforms, financial-grade systems, and large-scale data centers. Although existing observability frameworks provide abundant application-level and orchestration-level metrics, they often neglect the impact of low-level hardware behaviors on service reliability. In practice, PCIe protocol anomalies such as Completion Timeout, Replay Timer Expiration, and Link Retrain events may propagate across device topology and ultimately manifest as pod restarts, latency jitter, and SLA violations. To bridge this hardware–software visibility gap, we proposed PCGAT-Stab, an explainable Temporal Graph Attention Network that models PCIe protocol behavior and predicts system stability in cloud-native microservices through topology-aware and temporally adaptive learning. The proposed model constructs a dynamic PCIe topology graph to characterize device-level interactions and leverages multi-head graph attention to capture spatial propagation patterns of protocol errors. A recurrent temporal module further models cumulative and delayed effects of hardware anomalies across time windows. Through an explicit device–pod association mechanism, hardware-level embeddings are mapped to pod-level risk representations, enabling end-to-end instability prediction. In addition, attention weight analysis and SHAP-based attribution enhance interpretability, offering actionable diagnostic insights for DevOps and reliability engineering teams.

Experimental results on a real-world Kubernetes cluster dataset demonstrate that PCGAT-Stab achieves an AUC of 0.94 and an F1-score of 0.91 for instability classification, outperforming LSTM and Transformer baselines. For latency prediction, the model attains an RMSE of 5.72 and MAE of 4.68, confirming its superior capability in cross-layer modeling and temporal correlation learning. The training process converges stably within 50 epochs, with the final loss approaching 0.015, indicating effective optimization without significant overfitting. Despite these promising results, several limitations remain. The model is currently validated on a specific cluster scale and PCIe generation, and extreme failure scenarios or heterogeneous hardware environments require further evaluation. Specifically, drawing inspiration from collaborative fault diagnosis frameworks that successfully employ federated learning across distributed nodes to preserve data privacy [18], we plan to extend PCGAT-Stab to multi-cluster environments. This extension will facilitate joint stability modeling without requiring the centralization of sensitive, raw hardware telemetry. Future work may incorporate multi-cluster federated learning, integrate additional hardware telemetry such as NVLink and CXL, and explore self-supervised pretraining to enhance robustness. Future work may explore multi-cluster federated learning, integrate additional hardware telemetry such as NVLink and CXL, and explore self-supervised pretraining to enhance robustness. Furthermore, we aim to investigate the end-to-end impact of PCIe anomalies on specific downstream AI workloads, such as personalized financial advisory systems [19], to develop more application-aware stability metrics. Additionally, inspired by recent advancements in retrieval-augmented frameworks. Moreover, combining causal inference techniques with topology-aware graph modeling could further strengthen explainability and improve proactive fault prevention in next-generation cloud-native infrastructures. Future work may explore distributed training strategies and real-time inference optimization. Additionally, inspired by recent advancements in retrieval-augmented frameworks for industrial data warehouses [20], integrating Natural Language to SQL (NL2SQL) or RAG-based interfaces could provide a more intuitive way for operators to interact with our model's explainability components, enabling natural language queries over the identified hardware-to-service causal relationships.

References

- Schroeder, Bianca, and Garth A. Gibson. "Understanding failures in petascale computers." *Journal of Physics: Conference Series*. Vol. 78. No. 1. 2007.
- Gunawi, Haryadi S., et al. "What bugs live in the cloud? a study of 3000+ issues in cloud systems." *Proceedings of the ACM symposium on cloud computing*. 2014.
- Barroso, Luiz André, Urs Hölzle, and Parthasarathy Ranganathan. "Trusted Computing: Reliability, Availability, Security, Privacy." *The Data Center as a Computer: Designing Warehouse-Scale Machines*. Cham: Springer Nature Switzerland, 2025. 269-308.
- Todi, Shradha, and Pankaj Kumar Dubey. "A Software-based Diagnostic Approach to Detect PCIe Faults and Enhance the Reliability of Safety-critical Systems." 2023 IEEE Women in Technology Conference (WINTTECHCON). IEEE, 2023.
- Chen, Jian, et al. "TraceGra: A trace-based anomaly detection for microservice using graph deep learning." *Computer Communications* 204 (2023): 109-117.
- Soualhia, Mbarka, Chunyan Fu, and Foutse Khomh. "Infrastructure fault detection and prediction in edge cloud environments." *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. 2019.
- Allen, Sadie, et al. "Tritium: A cross-layer analytics system for enhancing microservice rollouts in the cloud." *Proceedings of the Seventh International Workshop on Container Technologies and Container Clouds*. 2021.

- Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).
- Velickovic, Petar, et al. "Graph attention networks." stat 1050.20 (2017): 10-48550.
- Zhang, Zhen, et al. "Towards optimized scheduling and allocation of heterogeneous resource via graph-enhanced EPSO algorithm." Journal of Cloud Computing 13.1 (2024): 108.
- Xu, Bo, Chenhao Li, and Lars Pettersson. "Attention-Based Graph Transformers for Fault-Tolerant Task Migration in Heterogeneous Data Centers." Journal of Computing and Electronic Information Management 19.1 (2025): 43-52.
- Wang, Tangtang, Kaijie Zhang, and Kuangcong Liu. "A knowledge graph and deep learning-based semantic recommendation database system for advertisement retrieval and personalization." Journal of Computer Science and Frontier Technologies 3.2 (2026): 79-91.
- Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." Advances in neural information processing systems 30 (2017).
- Wang, Yuyao, and Selahattin Sayil. "Soft Error Evaluation and Mitigation in Gate Diffusion Input Circuits." 2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS). IEEE, 2024.
- Liang, Zucheng, et al. "Research on multi-hop inference optimization of llm based on mquake framework." arXiv preprint arXiv:2509.04770 (2025).
- Huang, Yanxiang, et al. "Enhancing Open-Vocabulary Panoptic Segmentation with Semantic-Guided Q-Tuning." 2025 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2025.
- Yao, Yilin, Wenchao Zhang, and Mengtong Li. "Cloud-Edge Federated Incremental Learning Framework for PMSM Efficiency Optimization with Lightweight CNN-LSTM Models and OTA Differential Deployment." Journal of Computer Science and Frontier Technologies 3.1 (2026): 98-111.
- Zhang, Wenchao, Mengtong Li, and Xing Chen. "Distributed Edge-Fog-Cloud Architecture for Intelligent Fault Diagnosis of Permanent Magnet Synchronous Motors Using Temporal Fusion Transformers and Federated Learning." Fundamental Scientific Reports in Multidisciplinary Areas 2.02 (2026): 121-131.
- Jiang, Gaozhe, et al. "Investment advisory robotics 2.0: Leveraging deep neural networks for personalized financial guidance." (2025).
- Wei, Haoyan, Yuqing Wu, and Mengtong Li. "RAGN-IIoT: A Retrieval-Augmented NL2SQL Framework with Dynamic Sensor-Selection Guardrails for Industrial IoT Time-Series Data Warehouses." Journal of Computer, Signal, and System Research 2.7 (2025): 78-88.