RATIONANOMALY: LOG ANOMALY DETECTION WITH RATIONALITY VIA CHAIN-OF-THOUGHT AND REINFORCEMENT LEARNING

Song Xu[*†], *Yilun Liu*[†✉], *Minggui He*[†], *Mingchen Dai*[*†], *Ziang Chen*[§†],
Chunguang Zhao[†], *Jingzhou Du*[†], *Shimin Tao*[†], *Weibin Meng*[†],
Shenglin Zhang[§], *Yongqian Sun*[§], *Boxing Chen*[‡], *Daimeng Wei*[†]

[*]School of Software Engineering, University of Science and Technology of China, Hefei, China
[⋆]Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou, China
[†]Huawei, Beijing, China
[‡]Huawei Canada, Montreal, Canada
[§]Nankai University, Tianjin, China

ABSTRACT

Logs constitute a form of evidence signaling the operational status of software systems. Automated log anomaly detection is crucial for ensuring the reliability of modern software systems. However, existing approaches face significant limitations: traditional deep learning models lack interpretability and generalization, while methods leveraging Large Language Models are often hindered by unreliability and factual inaccuracies. To address these issues, we propose RationAnomaly, a novel framework that enhances log anomaly detection by synergizing Chain-of-Thought (CoT) fine-tuning with reinforcement learning. Our approach first instills expert-like reasoning patterns using CoT-guided supervised fine-tuning, grounded in a high-quality dataset corrected through a rigorous expert-driven process. Subsequently, a reinforcement learning phase with a multi-faceted reward function optimizes for accuracy and logical consistency, effectively mitigating hallucinations. Experimentally, RationAnomaly outperforms state-of-the-art baselines, achieving superior F1-scores on key benchmarks while providing transparent, step-by-step analytical outputs. We have released the corresponding resources, including code and datasets[1].

***Index Terms*—** Anomaly Detection, Fine-Tuning, Reinforcement Learning, Log Analysis, Large Language Model

1. INTRODUCTION

Logs represent a type of signal generated by software systems to attest to their operational state. Automated log analysis through machine learning has become essential for maintaining system reliability [1, 2]. However, existing approaches face significant limitations: traditional deep learning models lack interpretability and generalize poorly [3], while LLM-based methods are hindered by unreliability and factual inaccuracies [4]. Prompt-based LLM approaches suffer from hallucinations [5], and existing fine-tuning strategies fail to explicitly model step-by-step diagnostic reasoning [6].

To overcome the limitations, we introduce RationAnomaly, a novel framework that integrates Chain-of-Thought fine-tuning with reinforcement learning. Our approach begins with CoT-guided supervised fine-tuning on a high-quality, expert-validated dataset to instill systematic reasoning patterns. This is followed by a reinforcement learning phase employing a multi-faceted reward function that optimizes both detection accuracy and logical consistency, thereby effectively reducing hallucinations. Experiments demonstrate that RationAnomaly achieves state-of-the-art performance, delivering superior accuracy while offering transparent and interpretable analytical outputs.

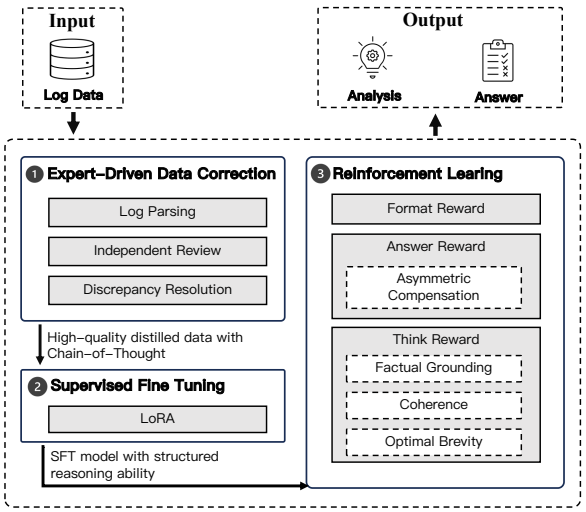

Fig. 1. Overview of RationAnomaly

✉ Corresponding author (liuyilun3@huawei.com).
[1]https://github.com/Gravityless/RationAnomaly

2. RELATED WORK

2.1. Deep Learning-based Log Anomaly Detection

There is a significant amount of research on applying deep learning to log anomaly detection [7, 8, 9, 10, 11]. DeepLog [12] used LSTMs to model normal patterns, while subsequent approaches enhanced performance with template embeddings (LogAnomaly [13]) and attention mechanisms (LogRobust [14]). Despite achieving high accuracy, these methods operate as black boxes, lacking interpretability.

2.2. LLM-based Log Anomaly Detection

Recent research leverages LLMs through prompt-based methods and fine-tuning approaches. Prompt-based methods include LogPrompt [15], EagerLog [16], LogRAG [17] and RAGLog [18], while fine-tuning methods include LogGPT [6], LogLM [5] and SuperLog [19]. These approaches either suffer from hallucinations or fail to explicitly model step-by-step reasoning.

Our work addresses this gap by explicitly optimizing the reasoning process through a two-stage paradigm: CoT-guided fine-tuning followed by reinforcement learning alignment, uniquely combining accuracy with interpretability.

3. METHODOLOGY

Our framework, RationAnomaly, enhances log anomaly detection through a multi-stage process, as illustrated in Fig. 1. The process begins with a foundational data correction stage to ensure data integrity, followed by a two-stage training paradigm: (1) **Chain-of-Thought Supervised Fine-Tuning (CoT-SFT)** to instill expert-like reasoning, and (2) **Reinforcement Learning Alignment (RLA)** to optimize for accuracy and reliability.

3.1. Expert-Driven Data Correction

The reliability of any data-driven model is contingent upon the quality of its training and evaluation data. Public benchmarks like BGL and Spirit, while widely used [20], contain systematic labeling errors that can compromise model evaluation. To establish a trustworthy foundation, we conducted a comprehensive data correction process.

We assembled a team of five industry experts to systematically review all 3,046 unique log templates extracted from the BGL and Spirit datasets. The process began with an independent review by each expert. Discrepancies were then resolved through consensus-driven panel discussions, with a senior expert providing final validation for disputed cases. As shown in Table 1, this rigorous process identified and corrected 225 systematically mislabeled log templates (7.4% of total). Our analysis revealed a significant bias towards false negatives (98.2% of errors), where critical system failures (e.g., "segmentation violation", "Connection refused") were incorrectly marked as normal. The result is a high-fidelity benchmark characterized by inter-annotator agreement at κ=0.94, forming a reliable basis for our subsequent training and evaluation.

3.2. Chain-of-Thought Supervised Fine-Tuning

The goal of this stage is to imbue the model with structured, reasoning capabilities before it delivers a final verdict. To achieve this, we leverage a powerful teacher model (GPT-4o) to distill a CoT dataset. For each log template in the training set, we prompted the teacher model to generate a step-by-step analysis that mimics an expert's diagnostic process: identifying key parameters, reasoning about their implications, and drawing a conclusion. This yields a high-quality dataset of (log, CoT-analysis, label) triplets. To maintain computational efficiency and prevent catastrophic forgetting of the base model's capabilities, we use Low-Rank Adaptation (LoRA) [21].

3.3. Reinforcement Learning Alignment

While SFT teaches reasoning patterns, it does not guarantee factual accuracy or reliability against hallucinations when it comes to various logs. The RLA stage addresses this by refining the model's behavior, aligning it with real-world operational goals through a meticulously designed reward function.

We employ Group Relative Policy Optimization (GRPO), an efficient and stable RL algorithm, to optimize the model. The cornerstone of this stage is our multi-faceted reward function, R_{total}, which provides a holistic evaluation of the model's generated output from three perspectives: format adherence, answer correctness, and reasoning quality.

Format Reward (R_{format}): A binary reward that is positive only if the output strictly adheres to the predefined structure (i.e., contains both ⟨think⟩ and ⟨answer⟩ sections). A failure here will result in the subsequent rewards being ignored, strongly discouraging malformed outputs.

Answer Reward (R_{answer}): To address the high cost of false negatives in production environments, and the unbalanced distribution between normal and abnormal classes in the dataset, we introduce an asymmetric reward mechanism: correctly identifying an anomaly results in a higher reward than correctly identifying a normal log, and missing an anomaly also leads to a heavier penalty.

Thinking Reward (R_{think}): To combat hallucination, the model's output is optimized along three crucial dimensions:

(1) Factual Grounding: Assesses the semantic overlap (using BLEU and ROUGE) between the generated analysis and the source content. This dimension ensures the reasoning is directly supported by evidence from the log, effectively discouraging hallucination.

Table 1. Detailed breakdown of annotation errors identified and corrected in the dataset.

Correction	Error Category	Count	Percentage	Representative Examples
Normal → Abnormal	System Error	78	34.7%	"PANIC: segmentation violation..." "hit ASSERT condition..."
	Network Issue	47	20.9%	"Connection refused..." "Bad UMNT RPC: RPC: Timed out"
	Hardware Failure	40	17.8%	"parity error in read queue..." "DDR failing info register..."
	Software Exception	56	24.9%	"ciod: Error loading..." "divide-by-zero..."
Abnormal → Normal	-	4	1.8%	"exited normally with exit code..." "Mounting NFS filesystems..."
Total	**-**	**225**	**100.0%**	**-**

(2) Coherence: Utilizes a perplexity model to evaluate the fluency and logical flow of the reasoning. This promotes outputs that are sensible and easy to follow, rather than repetitive or nonsensical.

(3) Optimal Brevity: Encourages concise yet complete explanations by assessing alignment with the target length derived from our distilled CoT dataset. This ensures the analysis is efficient without sacrificing critical information.

4. EXPERIMENTS

4.1. Experimental Setup

For the BGL and Spirit datasets, we performed a chronological split to simulate real-world data flow. To support deep learning baselines, we first sampled a 2000-log template-level training set (15% anomaly rate), then constructed session-level data using a 100-log fixed window [3]. Test sets were created similarly, resulting in 8000 entries and sessions per dataset. To prevent data leakage, logs used in RationAnomaly's training were excluded from these session-level test sets.

We compare RationAnomaly against two categories of state-of-the-art methods: conventional deep learning models, including unsupervised approaches like DeepLog [12] and LogAnomaly [13] as well as the supervised LogRobust [14]; and LLM-based techniques, such as LogPrompt [15] and a zero-shot Llama 2 7B baseline.

RationAnomaly is built upon Llama 2 7B and implemented using PyTorch, VeRL and Hugging Face libraries. All experiments were conducted on a server with 8x NVIDIA A100 GPUs. We use three standard metrics for evaluation: Precision, Recall, and F1-score.

4.2. Overall Performance

Table 2 presents a comprehensive performance comparison. RationAnomaly establishes a new state-of-the-art, achieving superior F1-scores across all evaluated datasets and scenarios.

Our method's performance surpasses that of conventional deep learning baselines. On session-level, **RationAnomaly achieves an F1-score of 0.958** on Spirit, a notable improvement over the best-performing baseline, LogAnomaly (0.925). Furthermore, unlike traditional methods which are limited to session-level analysis, RationAnomaly's semantic understanding enables effective template-level detection, achieving F1-scores of 0.887 on BGL and 0.862 on Spirit. This demonstrates its advanced capability for fine-grained log interpretation.

The performance delta between RationAnomaly and the zero-shot Llama 2 7B baseline highlights the necessity of our two-stage training paradigm. On the BGL template-level, **the F1-score shows a 29.3% relative improvement**. Moreover, a critical outcome of our method is the achievement of a **well-balanced precision-recall profile**. On the Spirit session dataset, it attains 0.959 for both precision and recall. This balance is a direct consequence of the Reinforcement Learning Alignment stage, which trains the model to be both accurate in its predictions and sensitive to genuine anomalies, enhancing its reliability for operational use.

4.3. Ablation Study

To dissect the contribution of each component within our framework, we conducted an ablation study. As shown in Fig. 2, the ablation experiment results show the same trend on BGL and Spirit datasets.

The model without CoT-SFT solely undergoes reinforcement learning achieves the lowest F1-score and significantly lower Precision and Recall values. This indicates that fine-

Table 2. Overall Result

Method	BGL(Session-level)			BGL(Template-level)			Spirit(Session-level)			Spirit(Template-level)		
	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec
DeepLog	0.869	0.811	0.935	-	-	-	0.905	0.891	0.919	-	-	-
LogAnomaly	0.857	0.770	**0.965**	-	-	-	0.925	0.880	0.975	-	-	-
LogRobust	0.811	0.704	0.956	-	-	-	0.921	0.856	**0.996**	-	-	-
LogPrompt	0.834	0.874	0.811	0.827	0.724	**0.964**	0.917	0.859	0.983	0.795	0.834	**0.858**
Llama 2 7B	0.707	0.741	0.753	0.686	0.858	0.633	0.800	0.881	0.861	0.655	0.839	0.598
RationAnomaly	**0.909**	**0.900**	0.919	**0.887**	**0.898**	0.881	**0.958**	**0.959**	0.959	**0.862**	**0.899**	0.847

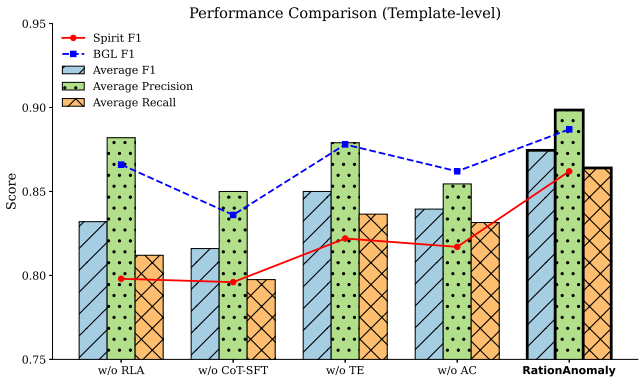

Fig. 2. Ablation Result

tuning on a structured reasoning process provides an important and effective foundation for subsequent steps.

The necessity of RLA stage is made evident by the significant performance increase from the model without RLA to the final RationAnomaly model, which is reflected in **F1-score improvement from 0.798 to 0.862** on Spirit. This result indicates that RLA is an essential step for refining the model's behavior, correcting subtle reasoning flaws, and aligning its outputs toward maximal accuracy.

The specific design of the reward function proves to be pivotal. Disabling the Asymmetric Compensation (without AC) or the Thinking Evaluation (without TE) leads to distinct performance degradation, with **F1-scores dropping from 0.862 to 0.817 and 0.822** on Spirit. This confirms that both applying greater penalties to high-cost errors and rewarding factually-grounded reasoning are critical mechanisms for achieving the model's final, robust performance.

4.4. Case Study

To demonstrate our model's capabilities, we present a case study in Fig. 3. The log entry requires domain-specific knowledge that challenges conventional methods. RationAnomaly not only correctly classifies it as abnormal but also generates a transparent, expert-like rationale. It leverages its acquired **Domain Knowledge** to define VPD, extracts the **Core Information** ("missing node"), and performs a **Rational Deduction** to identify a hardware detection failure. This structured output, a direct result of our CoT-SFT and RLA pipeline, showcases a shift from simple pattern matching to providing verifiable and trustworthy diagnostic insights.

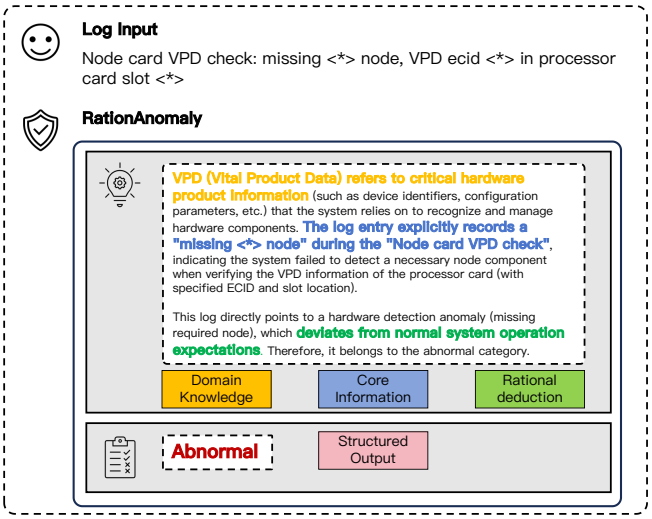

Fig. 3. A case study demonstrating RationAnomaly's step-by-step reasoning on a hardware-related log.

5. CONCLUSION

In this paper, we introduced RationAnomaly, a novel framework that significantly enhances the reliability and interpretability of log anomaly detection. Our two-stage paradigm, grounded in expert-corrected data, synergizes Chain-of-Thought fine-tuning with reinforcement learning alignment. Experiments demonstrate that RationAnomaly substantially outperforms existing methods, achieving superior accuracy while providing transparent, step-by-step reasoning. This work represents a significant step towards developing AIOps tools that are dependable and trustworthy, with future possibilities for extension into correlating multi-modal signals from logs, metrics, and traces.

6. REFERENCES

[1] S. Lin, R. Clark, R. Birke, S. Schönborn, N. Trigoni, and S. Roberts, "Anomaly detection for time series using vae-lstm hybrid model," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 4322–4326.

[2] L. Zhang, T. Jia, M. Jia, H. Liu, Y. Yang, Z. Wu, and Y. Li, "Towards close-to-zero runtime collection overhead: Raft-based anomaly diagnosis on system faults for distributed storage system," *IEEE Transactions on Services Computing*, vol. 18, no. 2, pp. 1054–1067, 2025.

[3] V. Le and H. Zhang, "Log-based anomaly detection with deep learning: How far are we?," *CoRR*, vol. abs/2202.04301, 2022.

[4] L. Zhang, T. Jia, M. Jia, Y. Wu, A. Liu, Y. Yang, Z. Wu, X. Hu, P. Yu, and Y. Li, "A survey of aiops in the era of large language models," *ACM Comput. Surv.*, vol. 58, no. 2, Sept. 2025.

[5] Y. Liu, Y. Ji, S. Tao, M. He, W. Meng, S. Zhang, Y. Sun, Y. Xie, B. Chen, and H. Yang, "Loglm: From task-based to instruction-based automated log analysis," *CoRR*, vol. abs/2410.09352, 2024.

[6] X. Han, S. Yuan, and M. Trabelsi, "Loggpt: Log anomaly detection via gpt," in *2023 IEEE International Conference on Big Data*, 2023, pp. 1117–1122.

[7] S. Nedelkoski, J. Bogatinovski, A. Acker, J. Cardoso, and O. Kao, "Self-attentive classification-based anomaly detection in unstructured logs," in *20th IEEE International Conference on Data Mining*. 2020, pp. 1196–1201, IEEE.

[8] L. Yang, J. Chen, Z. Wang, W. Wang, J. Jiang, X. Dong, and W. Zhang, "Plelog: Semi-supervised log-based anomaly detection via probabilistic label estimation," in *ICSE Companion 2021*. 2021, pp. 230–231, IEEE.

[9] X. Li, P. Chen, L. Jing, Z. He, and G. Yu, "Swisslog: Robust and unified deep learning based log anomaly detection for diverse faults," in *31st IEEE International Symposium on Software Reliability Engineering*. 2020, pp. 92–103, IEEE.

[10] T. Jia, Y. Wu, C. Hou, and Y. Li, "Logflash: Real-time streaming anomaly detection and diagnosis from system logs for large-scale software systems," in *2021 IEEE 32nd ISSRE*, 2021, pp. 80–90.

[11] X. Xie, S. Jian, C. Huang, F. Yu, and Y. Deng, "Logrep: Log-based anomaly detection by representing both semantic and numeric information in raw messages," in *2023 IEEE 34th ISSRE*, 2023, pp. 194–206.

[12] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017, pp. 1285–1298, ACM.

[13] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, Y. Chen, R. Zhang, S. Tao, P. Sun, and R. Zhou, "Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. 2019, pp. 4739–4745, ijcai.org.

[14] X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. Li, J. Chen, X. He, R. Yao, J. Lou, M. Chintalapati, F. Shen, and D. Zhang, "Robust log-based anomaly detection on unstable log data," in *ESEC/SIGSOFT FSE 2019*. 2019, pp. 807–817, ACM.

[15] Y. Liu, S. Tao, W. Meng, J. Wang, W. Ma, Y. Chen, Y. Zhao, H. Yang, and Y. Jiang, "Interpretable online log analysis using large language models with prompt strategies," in *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension*. 2024, pp. 35–46, ACM.

[16] C. Duan, T. Jia, Y. Yang, G. Liu, J. Liu, H. Zhang, Q. Zhou, Y. Li, and G. Huang, "Eagerlog: Active learning enhanced retrieval augmented generation for log-based anomaly detection," in *ICASSP 2025 - 2025 IEEE ICASSP*, 2025, pp. 1–5.

[17] W. Zhang, Q. Zhang, E. Yu, Y. Ren, Y. Meng, M. Qiu, and J. Wang, "Lograg: Semi-supervised log-based anomaly detection with retrieval-augmented generation," in *IEEE International Conference on Web Services*. 2024, pp. 1100–1102, IEEE.

[18] J. Pan, S. L. Wong, and Y. Yuan, "Raglog: Log anomaly detection using retrieval augmented generation," *CoRR*, vol. abs/2311.05261, 2023.

[19] Y. Ji, Y. Liu, F. Yao, M. He, S. Tao, X. Zhao, C. Su, X. Yang, W. Meng, Y. Xie, B. Chen, and H. Yang, "Adapting large language models to log analysis with interpretable domain knowledge," *CoRR*, vol. abs/2412.01377, 2024.

[20] S. He, J. Zhu, P. He, and M. R. Lyu, "Loghub: A large collection of system log datasets towards automated log analytics," *CoRR*, vol. abs/2008.06448, 2020.

[21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *CoRR*, vol. abs/2106.09685, 2021.