DynRCA: Dynamic Sampling for Efficient Multi-Agent Reasoning in Microservice RCA

Yuke Liu*, Yitao Xiao*, Guoming Yang*, Shaoyong Guo†, Feng Qi†
*Beijing University of Posts and Telecommunications, Beijing, China
†State Key Lab of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
Email: 2024140887@bupt.cn, Yitoooom@bupt.edu.cn, gmyang@bupt.edu.cn,
syguo@bupt.edu.cn, qifeng@bupt.edu.cn

***Abstract*—The intricate invocation dependencies and fault cascades within microservice architectures pose significant challenges for Large Language Model (LLM)-based Root Cause Analysis (RCA). Prevailing methodologies are often constrained by static sampling, which fail to adapt reasoning width to fault complexity, leading to redundant trajectory exploration and accumulation of decisional noise. To address these limitations, we propose DynRCA, a multi-agent collaborative framework featuring adaptive reasoning trajectory optimization tailored for root cause analysis of microservice environments. DynRCA introduces a dynamic width sampling mechanism that leverages high-dimensional embeddings and density-based clustering to identify and consolidate redundant reasoning steps in real-time. This approach effectively prunes the search space while preserving diverse, high-value diagnostic paths. Experimental evaluations on the 2025 AIOps Challenge and RE2-OnlineBoutique datasets demonstrate that DynRCA achieves high accuracy of 64.25% while reducing reasoning overhead by 28.5%, striking an optimal balance between diagnostic depth and computational efficiency.**

***Index Terms*—Root Cause Analysis, Large Language Models, Multi-Agent System, Dynamic Sampling**

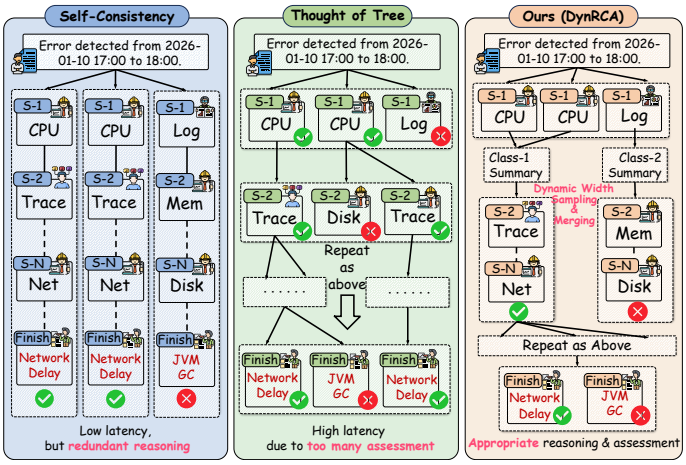


Fig. 1. Comparison of different sampling strategies. SC exhibits excessive path homogeneity, while ToT suffers from reduced reasoning speed due to its rigorous layer-wise verification. DynRCA effectively addresses these issues.

I. Introduction

Rapidly evolving cloud computing and containerization established microservice architecture as the prevailing paradigm for large-scale Internet of Things (IoT) applications [1]–[3]. By decoupling monolithic systems into fine-grained services, this architectural style enhances developmental agility and scalability. However, such decentralization inherently escalates operational complexity. Intricate, dynamic dependencies render individual node failures susceptible to cascading effects along call chains, potentially triggering service avalanches [4]. Faced with multimodal observability data, such as metrics, logs, and traces, Site Reliability Engineers (SREs) must execute complex decisions, from anomaly detection to root cause analysis, within stringent time windows. This poses formidable challenges to the precision and timeliness of Artificial Intelligence for IT Operations (AIOps).

Traditional RCA methods based on rules [5], [6] or deep learning [7]–[12] exhibit inherent limitations in processing unstructured data. Recently, Large Language Models (LLMs) have emerged as the central RCA paradigm due to their powerful reasoning and planning capabilities [13]–[16]. Early single-agent efforts primarily followed two trajectories: injecting historical failure patterns via fine-tuning [17], [18] and mitigating hallucinations through Retrieval-Augmented Generation (RAG) [19], [20]. However, massive microservice operational data significantly exceeds single-agent context windows, hindering the resolution of complex faults involving long-range dependencies. Furthermore, lacking self-correction mechanisms, these methods suffer from noise accumulation across reasoning chains, resulting in insufficient robustness.

To transcend single-agent performance bottlenecks, multi-agent collaborative frameworks exhibit superiority in complex fault localization via role specialization and interaction mechanisms [21]. For instance, Flow-of-Action [22] transforms Standard Operating Procedures (SOPs) into executable code constraints, whereas mABC [23] employs blockchain-inspired consensus to audit reasoning outcomes. Nevertheless, contemporary approaches face persistent challenges. Static SOPs fail to generalize to non-deterministic Out-of-Distribution (OOD) faults, while greedy or rigid voting mechanisms are prone to infinite loops, hindering high-confidence consensus within deterministic latency constraints.

To augment the precision of agent-based Root Cause Anal-

This work was supported by the National Natural Science Foundation of China (62322103).
*Corresponding author: Shaoyong Guo.

ysis (RCA), contemporary frameworks extensively integrate Chain-of-Thought (CoT) [24] and its derivatives, including CoT-SC, ToT, and Beam Search [25]–[27], to mitigate reasoning error propagation via multi-path sampling and aggregation. A notable advancement, RCAgent [28], employs Trajectory-level Self-Consistency (TSC) to suppress early-stage redundancy by deferring the sampling onset. However, as illustrated in Fig. 1, these paradigms remain constrained by rigid sampling mechanisms that fail to dynamically modulate sampling width in accordance with diagnostic complexity. This inability to adaptively allocate resources results in significant latency from redundant reasoning and lacks efficient intermediate state aggregation. Consequently, erroneous information persists unintercepted across multi-round dialogues, ultimately leading to cascading error propagation.

To address these limitations, we propose DynRCA, an efficient multi-agent dynamic sampling reasoning method for microservice root cause analysis. DynRCA adaptively regulates reasoning sampling width via semantic clustering and classifier guidance, complemented by trajectory aggregation at critical nodes. This framework compresses computational overhead and latency while maintaining superior diagnostic precision. The primary contributions of this work are as follows:

- **The DynRCA multi-agent collaborative framework:** For microservice RCA, we developed agents encompassing task planning, multi-modal expert analysis, and arbitration evaluation. By decoupling tasks and specializing roles, this framework enhances orchestration flexibility and collaborative diagnostic capabilities within multi-agent RCA systems facing complex fault cascades.
- **Dynamic width sampling and unsupervised trajectory merging mechanism:** We developed a dynamic sampling mechanism to achieve adaptive reasoning trajectory adjustment. To overcome the scarcity of expert-labeled data, we propose an unsupervised trajectory classification strategy using UMAP-HDBSCAN for pseudo-labeling and GCE loss for classifier training. This approach effectively merges redundant exploration branches and suppresses error accumulation in noisy environments.
- **Experimental Validation:** Extensive comparative experiments and ablation studies were conducted using two public datasets. The results demonstrate that DynRCA achieves robust root cause localization accuracy of 64.25% while delivering 28.5% enhancement in average reasoning efficiency, highlighting its effectiveness in balancing performance and resource consumption.

II. Related Works

A. Root Cause Analysis

Root Cause Analysis aims to precisely pinpoint the origins of anomalies within complex failure symptoms, which is vital for ensuring microservice availability. As system complexity scales, RCA techniques have evolved from rule-based [5] to deep learning [12], and recently toward LLM-based approaches that leverage superior reasoning and tool-use capabilities. [14] pioneered the exploration of ReAct agents, which dynamically invoke database and knowledge-base retrieval tools to obtain real-time observability data for automated diagnosis. COCA [20] indexes function-level code snippets to reconstruct execution paths, effectively mitigating performance degradation in scenarios with missing observability data. Nevertheless, the massive data in microservices often exceeds the context window capacity of singular agents, leading to information truncation and context loss during long-chain reasoning for complex faults. Furthermore, the absence of self-correction mechanisms makes reasoning chains susceptible to noise accumulation, compromising model robustness.

B. LLM Multi-Agent

Multi-agent frameworks are designed to overcome monolithic limitations regarding context windows and noise accumulation in extended reasoning chains. Current collaborative paradigms, such as Chain [29], Debate [30] and DyLAN [31], effectively surpass performance bottlenecks by reducing perceptual load and enhancing fault tolerance. Building on this basis, more diversified multi-agent collaborative frameworks have been developed. Flow-of-Action [22] integrates SOPs into knowledge base and transforms them into executable toolchains, imposing hard constraints at critical nodes to ensure the compliance of reasoning trajectories. However, rigid SOPs curtail the generalization of reasoning, which limits agents' ability to deal with unexpected faults. mABC [23] proposes a comprehensive multi-agent workflow ranging from alert triggering to resolution generation, incorporating a consensus mechanism for weighted voting and review of reasoning steps. Despite these advancements, greedy-strategy reasoning and rigid voting mechanisms struggle to achieve high confidence consensus within limited steps, often leading to logical deadlocks and failing to satisfy the stringent requirements for deterministic latency in RCA.

C. Agent Reasoning Trajectories Sampling

Agent reasoning trajectory sampling involves the filtering of intermediate decision steps and interaction sequences generated during multi-agent collaboration. By preserving diverse, high-value reasoning paths, this technique aims to bolster agent adaptability to OOD fault diagnosis, thereby extending generalization boundaries and enhancing system stability in unpredictable environments. Beam Search [26] maintains multiple candidate sequences and prunes low quality branches. Although it expands the search space, iterative evaluation at each step incurs substantial computational latency. SC [25] samples multiple reasoning trajectories in parallel and employs majority voting to enhance reliability, yet its stochastic sampling prone to errors and redundancy. RCAgent [28] refines this via TSC, which delays sampling to the final reasoning stage to share initial steps and optimize costs. Nonetheless, similar to SC, TSC remains constrained by fixed sampling width and requires all branches to complete before aggregation, ultimately failing to fundamentally intercept cascading propagation of erroneous reasoning within branches.

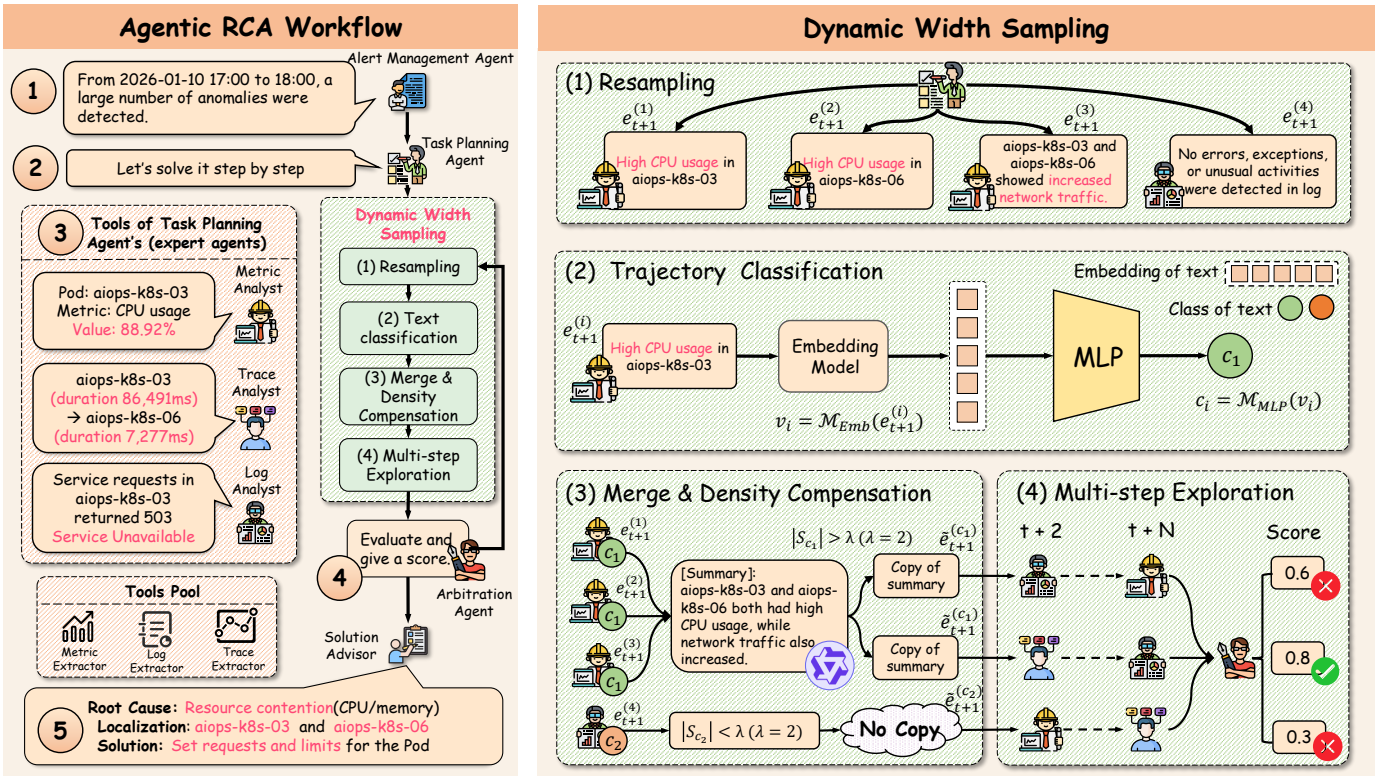


Fig. 2. Overview of DynRCA. The left side shows agent's workflow and possible outputs, while the right side shows the details of dynamic width sampling.

III. Method

A. Overview

We propose DynRCA, a collaborative multi-agent framework tailored for root cause analysis in microservice scenarios as illustrated in Fig. 2. Comprising seven specialized agents, DynRCA achieves efficient and automated RCA from raw alerts to solution generation through arbitration evaluation and reasoning path consolidation. The workflow of DynRCA is defined by the following phases:

*1) **Alert Pre-processing**:* As the entry point of the workflow, the Alert Management Agent is responsible for aggregating raw alerts generated by the microservice system. It performs priority assessment based on features including alert type, duration, and scope of influence before populating a queue with structured event reports.

*2) **Task Orchestration and Scheduling**:* Task Planning Agent extracts tasks from the queue to act as the central control hub. It leverages the ReAct [32] paradigm to drive the entire analytical process and dispatches instructions to underlying expert agents on demand.

*3) **Multi-dimensional Expert Analysis**:* During this stage, Metric, Log, and Trace Analyst receive planning instructions to invoke specific data acquisition tools. These agents extract anomalous features from heterogeneous data and transform them into textual summaries for planning layer.

*4) **Reasoning Trajectory Optimization**:* To enhance analytical efficiency, the system implements reasoning trajectory optimization. The Arbitration Agent utilizes CoT to quantitatively score candidate reasoning trajectories generated from expert feedback. By evaluating logical consistency and conflicting indicators, the agent identifies the optimal trajectory for integration into the context of the Task Planning Agent, facilitating dynamic correction of reasoning paths and elimination of redundancies. Steps *2)*, *3)*, and *4)* are executed iteratively until the Task Planning Agent determines that root cause localization is complete.

*5) **Conclusion and Solution Generation**:* After identifying the root cause, Solution Advisor synthesizes full context of reasoning evidence to output the final RCA conclusion alongside corresponding remediation recommendations.

B. Dynamic Width Sampling for Reasoning Trajectories

In the phase of reasoning trajectory optimization, we propose a dynamic width sampling method specifically designed for reasoning trajectories. This approach guides the reasoning and tool-invocation behaviors of the Task Planning Agent through semantic clustering and systematic trajectory evaluation, thereby balancing reasoning depth and computational overhead in complex microservice RCA tasks.

The reasoning process of the Task Planning Agent is modeled as a Markov Decision Process (MDP). Let H_t denote the historical reasoning trajectory up to time t. To capture the comprehensive context of decision-making, each reasoning step e_t is defined as a semantic triplet:

$$e_t = \left(e_t^{tht}, e_t^{act}, e_t^{obs}\right) \tag{1}$$

where e_t^{tht} represents the internal CoT of the agent, e_t^{act} denotes the action commands issued to specific expert agents, and e_t^{obs} signifies the summarized feedback returned by those experts. The historical trajectory is represented as:

$$H_t = \{e_0, e_1, \ldots, e_t\} \tag{2}$$

At time $t+1$, to avoid the limitations of singular reasoning path, Task Planning Agent executes k parallel stochastic samplings based on H_t, generating a set of candidate reasoning steps $E_{t+1} = \{e_{t+1}^{(1)}, \ldots, e_{t+1}^{(k)}\}$. Given that raw text sequences often exhibit high sparsity, direct deduplication within the textual space tends to overlook underlying semantic correlations. Consequently, the discrete text sequences are first projected into a high-dimensional latent space $\mathbb{R}^d$. A pre-trained embedding model $\mathcal{M}_{Emb}$ is employed to extract feature vectors v_i for each candidate step:

$$v_i = \mathcal{M}_{Emb}(e_{t+1}^{(i)}), \quad i \in \{1, \ldots, k\} \tag{3}$$

Next, a reasoning trajectory classifier $\mathcal{M}_{MLP}$ performs logical clustering on these semantic vectors. This stage aims to group reasoning paths that are semantically analogous and only redundant in expression into the same category c_i:

$$c_i = \mathcal{M}_{MLP}(v_i) \tag{4}$$

Upon obtaining the categorical partitions, for each subset $S_c = \{e_{t+1}^{(i)} \mid c_i = c\}$ belonging to category c, we utilize the inductive capabilities of LLM for conflict resolution and knowledge extraction. This process synthesizes a more representative and concise candidate step $\tilde{e}_{t+1}^{(c)}$ by eliminating logical contradictions between different sampling branches:

$$\tilde{e}_{t+1}^{(c)} = \mathcal{M}_{LLM}(S_c) \tag{5}$$

However, simply merging would erase the frequency information inherent in original sampling, thereby diminishing the weight of majority opinions. To calibrate the distribution of reasoning weights, we introduce a density compensation threshold λ. If the size of the original subset $|S_c|$ exceeds λ, indicating high consensus for that reasoning direction, we fix its representation in subsequent searches to λ via redundant replication; otherwise, a single sample is retained. The final calibrated candidate set $\tilde{E}_{t+1}$ is formulated as follows:

$$\tilde{E}_{t+1} = \bigcup_{c \in C} \begin{cases} \{\tilde{e}_{t+1}^{(c)}\} \times \lambda & \text{if } |S_c| > \lambda \\ \{\tilde{e}_{t+1}^{(c)}\} & \text{if } |S_c| \leq \lambda \end{cases} \tag{6}$$

To evaluate the long-term efficacy of current decisions, the system performs look-ahead exploration for each candidate branch in $\tilde{E}_{t+1}$. Starting from $\tilde{e}_{t+1}^{(c)}$, the agent continues reasoning for n steps until a predefined maximum search depth N is reached or a termination token is triggered. This procedure generates the set of candidate trajectories $\mathcal{T}$:

$$\begin{aligned} \tau^{(c)} &= \{\tilde{e}_{t+1}^{(c)}, (e_j)_{j=t+2}^{t+n}\}, \\ \text{s.t. } n &= \min\{j \mid e_j = \text{finish} \vee j = N\} \end{aligned} \tag{7}$$

Such look-ahead exploration facilitates the identification of deceptive paths that are locally optimal yet globally infeasible.

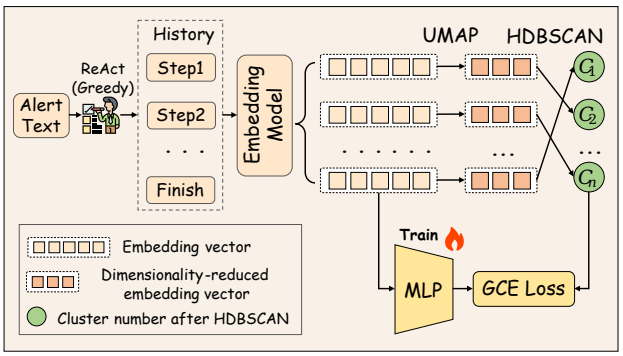


Fig. 3. The training process of reasoning trajectory classifier, encompassing dataset construction and model optimization.

Finally, the Arbitration Agent serves as a global observer to score the trajectories in $\mathcal{T}$ based on content consistency and the logical integrity of the evidence chain. The scoring function $\mathcal{R}(\cdot)$ evaluates whether the reasoning steps align logically with the original alert evidence, yielding a value in the range $(0, 1]$. The system selects the candidate trajectory τ^* with the highest score to update the historical trajectory:

$$\tau^* = \arg\max_{\tau \in \mathcal{T}} \mathcal{R}(\tau) \tag{8}$$

$$H_{t+n} = H_t \cup \{\tau^*\} \tag{9}$$

Following the update of historical trajectory, the dynamic sampling cycle is repeated until the arbitration agent selects a trajectory that terminates with Finish state.

C. Design of the Reasoning Trajectory Classifier

In the dynamic sampling process in Section III-B, the performance of the reasoning trajectory cluster governs the precision of reasoning path consolidation. While supervised classification excels in semantic modeling, its effectiveness in microservice root cause analysis is constrained by the intricate domain expertise and extended logical chains required, which render large-scale manual annotation cost-prohibitive. Consequently, we implement an unsupervised label generation method based on a cluster-and-map strategy, as shown in Fig. 3. This approach is utilized to train a Multi-Layer Perceptron (MLP) to execute clustering on reasoning paths.

To obtain domain-specific supervisory signals without human intervention, we construct the training set from alert data through an automated pipeline as follows:

*1) **Raw Corpora Generation**:* Initially, event reports produced by the Alert Management Agent serve as primary input. The Task-Planning Agent generates complete reasoning trajectories via ReAct paradigm under greedy sampling strategy. Each reasoning triplet e_t within the trajectory is treated as an independent textual unit and encoded into high-dimensional semantic vector using a pre-trained embedding model.

*2) **Manifold Alignment and Dimensionality Reduction**:* High-dimensional semantic spaces often exhibit significant sparsity, which lead to the degradation of distance metrics during direct clustering. We first apply L_2 normalization to the semantic vectors to eliminate magnitude-induced bias.

Subsequently, we introduce Uniform Manifold Approximation and Projection (UMAP) [33]. Grounded in manifold learning theory, UMAP constructs a topological graph structure to project high-dimensional features into a lower-dimensional space $\mathbb{R}^k$, effectively preserving both the local neighborhood structure and the global topological characteristics of the original semantic distribution.

*3) **Adaptive Density-Based Clustering**:* Given that the distribution of unstructured text vectors is difficult to ascertain a priori, we employ HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [34]. This method balances clustering depth by measuring the stability of clusters across varying density scales, thereby adaptively determining the number of clusters C without requiring predefined cluster centers.

- Vectors satisfying the density requirements are assigned to their respective cluster indices.
- Outliers that fail to meet the clustering thresholds are categorized into a dedicated noise class, thereby sharpening the discriminative boundaries of the classifier.
- The resulting feature-cluster pairs (v_i, c_i) serve as pseudo-labels, providing supervisory signals for the subsequent training of the MLP.

Owing to the inherent information loss during UMAP-based dimensionality reduction and the potential introduction of inaccurate pseudo-labels by the clustering algorithm, the training set inevitably contains a non-negligible degree of labeling noise. To enhance the model's robustness against such noise, we eschew the conventional Cross Entropy (CE) loss and instead adopt the Generalized Cross Entropy (GCE) [35] loss function to supervise the training of the MLP:

$$L = \sum_{i=1}^{C} \frac{1 - (y_i . \hat{y}_i)^q}{q} \tag{10}$$

Where C denotes the total number of logical categories determined through the clustering phase, y_i represents the one-hot encoded vector of the assigned pseudo-labels, $\hat{y_i}$ signifies the predicted probability distribution of the model following Softmax activation, $q \in (0, 1]$ serves as a tunable hyperparameter that modulates the trade-off between noise robustness and convergence speed.

IV. Experiment

A. Experiment Settings

*1) **Datasets**:* To evaluate the effectiveness of DynRCA, we utilize two benchmark datasets: the 2025 AIOps International Challenge dataset (2025 AIOps) and the RE2-OnlineBoutique (OB) dataset from the RCAEval benchmark [36]. 2025 AIOps is specifically curated to detect alert events and identify root causes within microservice architectures. It comprises 400 authentic failure cases, encompassing 15 distinct fault types across three hierarchical levels: node, pod, and service. Derived from the RCAEval benchmark, RE2-OB includes 90 fault injection scenarios. It covers 5 critical services and 6 representative fault categories.

*2) **Model Settings and Tools Design**:* For reasoning, we utilize Qwen3-8B, Llama-3.1-8B-Instruct, and GPT-4o-mini. The temperature is fixed at 0.5 to balance stability and creativity. For text embeddings, Qwen3-Embedding-8B is employed with an output dimensionality of 1024. To handle the multi-source data datasets, we developed specialized tools for time-specific information retrieval and redundant field filtering. All baselines utilize those tools to ensure experimental fairness.

*3) **Baselines**:* We chose SC, ToT with beam search and mABC as baselines to compare DynRCA. Specifically, the aggregation processes for SC and the evaluation phase of ToT are orchestrated by the Arbitration Agent.

*4) **Hyper Parameters**:* For dynamic sampling, we set the random sampling width to 8, the density compensation threshold λ to 3, and the maximum look-ahead search depth N to 5. Regarding dataset construction, UMAP parameters are configured with $n\_neighbors = 15$, $min\_dist = 0.1$, and target dimensionality of 10; meanwhile, HDBSCAN is parameterized with a minimum cluster size of 5 and minimum samples of 10. The q in the loss function is specified as 0.7. For baselines, the sampling widths for SC and Beam Search are consistently set to 8, with the beam width fixed at 4.

*5) **Evaluation Metric**:* To evaluate the effectiveness of DynRCA and baselines, we measure following two metrics:

- $Accuracy$: This metric measures whether the predicted root cause component set is identical to the ground truth. Any discrepancy, such as missing or extraneous components, results in a score of 0.
- $Top-k (k = 3, 5)$: This assesses whether the ground truth components are fully contained within the top-k most likely root causes generated by the model. For this metric, the model is prompted to produce k candidate root cause components ranked by confidence.

B. Main Results

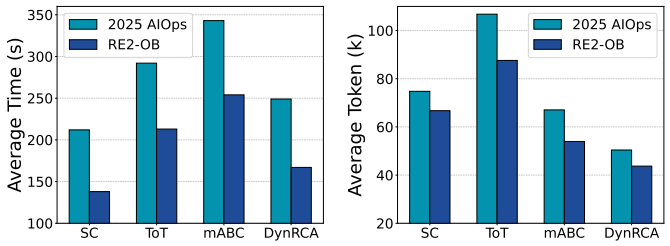
*1) **Performance Analysis**:* Table I presents a comparative analysis of diagnostic $Accuracy$ and $Top-k$ metrics between DynRCA and several baselines across two datasets.

On the 2025 AIOps dataset, DynRCA (powered by GPT-4o-mini) achieved an $Accuracy$ of 64.25%, marking a 6.5% improvement over the mABC baseline. Notably, DynRCA achieved state-of-the-art (SOTA) performance across all metrics on the RE2-OB dataset, demonstrating an accuracy enhancement ranging from 4.45% to 10.00% compared to all evaluated baselines. Conversely, the SC approach exhibited the most inferior performance, with accuracies limited to 47.75% and 36.67% on the two respective datasets.

The performance degradation of SC stems from the absence of a verification mechanism for reasoning paths, which renders the sampled branches susceptible to irreversible semantic noise and leads to cumulative deviation from the ground truth. While ToT and mABC incorporate evaluative components, their limited look-ahead capability restricts them to step-wise local assessments. This lack of a global perspective over multi-step search spaces often results in the premature pruning of optimal solutions in complex tasks. In contrast, the proposed

TABLE I
PERFORMANCE COMPARISON WITH DIFFERENT BASELINES ON THE TWO RCA DATASETS.

Method	Model	AIOps2025			RCAEval		
		Accuracy	Top-3	Top-5	Accuracy	Top-3	Top-5
SC	Qwen3-8B	45.75%	64.50%	68.75%	33.33%	53.33%	62.22%
	GPT-4o-mini	47.75%	62.75%	68.50%	36.67%	57.78%	65.56%
	Llama-3.1-8B-Instruct	43.50%	57.25%	62.25%	31.11%	46.67%	53.33%
ToT	Qwen3-8B	55.25%	67.00%	76.25%	41.11%	67.78%	77.78%
	GPT-4o-mini	56.00%	67.50%	**78.75%**	42.22%	66.67%	**78.89%**
	Llama-3.1-8B-Instruct	53.00%	66.75%	70.25%	37.78%	63.33%	71.11%
mABC	Qwen3-8B	55.75%	67.50%	74.25%	40.00%	67.78%	74.44%
	GPT-4o-mini	57.75%	68.00%	74.75%	41.11%	64.44%	77.78%
	Llama-3.1-8B-Instruct	54.50%	64.50%	68.75%	36.67%	60.0%	71.11%
DynRCA	Qwen3-8B	62.75%	67.75%	76.00%	43.33%	**68.89%**	75.56%
	GPT-4o-mini	**64.25%**	**70.50%**	77.25%	**46.67%**	64.44%	**78.89%**
	Llama-3.1-8B-Instruct	57.25%	65.75%	72.50%	38.89%	63.33%	73.33%



(a) Time Consumption (b) Token Consumption

Fig. 4. Efficiency Analysis of Different Methods

DynRCA framework integrates adaptive review timing with long-horizon reasoning steps, effectively traversing high-value trajectories and mitigating the risk of discarding global optima during root cause analysis.

*2) **Efficiency Analysis**:* We evaluate the efficiency of DynRCA (Qwen3-8B) in terms of temporal latency and token expenditure, as illustrated in Fig. 4.

Regarding execution speed, SC demonstrates lowest latency (212s and 138s) due to its simple architecture with single-phase aggregation. In contrast, mABC employs a fine-grained, step-wise verification mechanism that mandates evaluation by all agents at every decision node, triggering regeneration whenever consensus is not reached. This high-frequency interaction leads to substantial reasoning latency (343s and 254s). DynRCA mitigates this by optimizing evaluation frequency, maintaining competitive execution times of 249s and 167s. Although DynRCA exhibits a marginally lower inference speed than SC due to additional evaluation cycles, it provides superior diagnostic gains, achieving an optimal trade-off between inference efficiency and accuracy.

In terms of token consumption, the collective agent review mechanism of mABC leads to notably high resource requirements (74.7k and 66.7k tokens). ToT reaches peak consumption levels (106.8k and 87.6k tokens) owing to its fixed-width parallel sampling and incessant evaluative operations. DynRCA effectively addresses this by consolidating redundant reasoning trajectories post-sampling. Consequently, the token consumption of DynRCA is reduced to only 50.4k and 43.7k tokens, representing a substantial 28.5% decrease compared to mABC. This approach significantly enhances resource efficiency by pruning redundant reasoning paths and implementing periodic verification mechanisms.

C. Ablation Study

We conduct an ablation study using three variants of DynRCA (Qwen3-8B). As shown in Table II, the full configuration of DynRCA achieves superior performance across all metrics on both datasets, demonstrating the indispensable synergy of its constituent components.

*1) **Necessity of Multi-Agent Collaboration**:* Restricting the framework to a single-agent configuration leads to a substantial performance degradation of 25.00%. This decline underscores the pivotal role of multi-agent coordination in decomposing complex root cause analysis tasks, which single-agent architectures struggle to manage effectively.

*2) **Indispensability of External Tools**:* The exclusion of tool integration forces the model into a pure context-based reasoning mode. Due to the introduction of excessive irrelevant data, performance plummets to a baseline of 17.25%. This result highlights that tool utilization is critical for filtering environmental noise and extracting salient failure features.

*3) **Gains from Dynamic Width Sampling**:* Removing the dynamic width sampling mechanism also adversely affects performance. Although the impact is relatively less severe than that of the previous two components, this mechanism

TABLE II
ABLATION STUDY OF DYNRCA COMPONENTS.

Method	AIOps2025			RCAEval		
	Accuracy	Top-3	Top-5	Accuracy	Top-3	Top-5
DynRCA(Qwen3-8B)	62.75%	67.75%	76.00%	43.33%	68.89%	75.56%
w/o Multi-Agent	37.75%	52.00%	60.50%	27.78%	38.89%	52.22%
w/o Dynamic Sampling Mechanism	40.50%	58.75%	63.33%	34.44%	45.56%	58.89%
w/o Tools	17.25%	24.25%	30.75%	21.11%	32.22%	40.00%

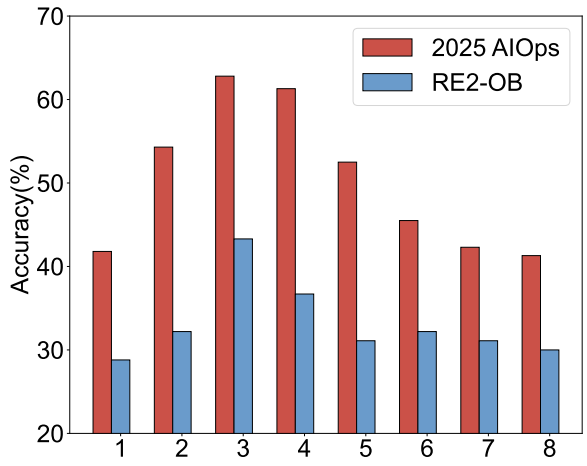


Fig. 5. Sensitivity Analysis of Density Compensation Threshold

is nonetheless essential for expanding the performance ceiling of RCA by optimizing reasoning trajectories.

D. Sensitivity Analysis

We evaluated the impact of the density compensation threshold on the *Accuracy* of DynRCA (Qwen3-8B) using the AIOps 2025 dataset, as illustrated in Fig. 5 The results indicate that the *Accuracy* undergoes a rapid initial increase followed by a sharp decline as the threshold rises, eventually stabilizing after the threshold exceeds 6. The *Accuracy* reaches its nadir at 41.25% and 28.89% on the two datasets, respectively. This performance degradation is primarily attributed to the excessive consolidation of reasoning trajectories. Specifically, when the threshold is set too low or too high, similar reasoning paths are over-merged, which disproportionately dilutes their significance during the evaluation phase and prevents optimal paths from being selected.

E. Case Study

In a representative case from the 2025 AIOps dataset, DynRCA demonstrates architectural superiority. Through its dynamic sampling and merging mechanism, the framework identifies analogous CPU anomalies across nodes and consolidates them into a single inference branch, eliminating computational redundancy. During evaluation, it prune ineffective paths (e.g. Branch-3), focusing reasoning on high-value trajectories. This coordination allows the Task Planning Agent

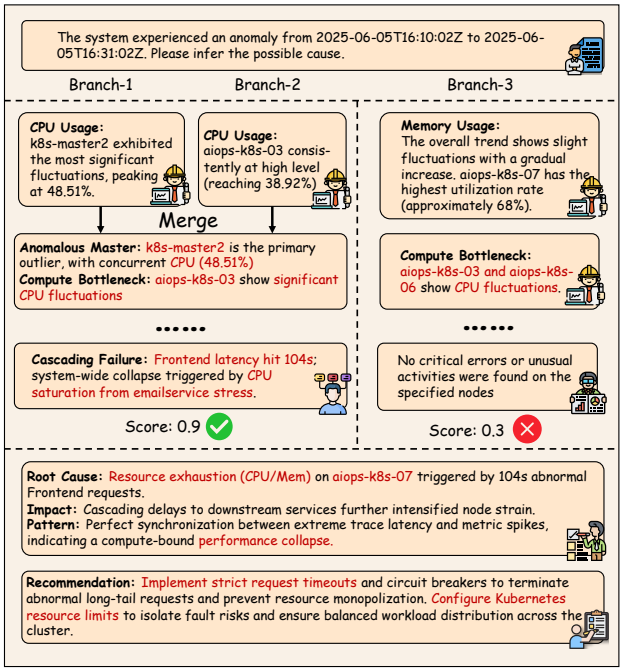


Fig. 6. Case of DynRCA in 2025 AIOps

to bypass noise and precisely isolate the root node aiops-k8s-07 and its underlying resource exhaustion. Furthermore, the integrated Solution Advisor facilitates a robust, closed-loop process from localization to actionable remediation.

V. CONCLUSION

In this work, we propose DynRCA, a multi-agent framework specifically designed for root cause analysis tasks in complex microservice scenarios. By introducing a semantic-based dynamic width sampling mechanism, DynRCA effectively alleviates the redundancy and noise issues inherent in static sampling strategies. Our approach utilizes high-dimensional embeddings and density-based clustering algorithms to adaptively prune reasoning trajectories, effectively compressing the search space while preserving critical diagnostic paths. Extensive experiments on multiple datasets demonstrate that DynRCA can balance the diagnostic depth and computational overhead of RCA, significantly reducing diagnostic duration and thereby improving automated operations and maintenance capabilities for microservices.

REFERENCES

[1] M. Ma, Y. Liu, Y. Tong, H. Li, P. Zhao, Y. Xu, H. Zhang, S. He, L. Wang, Y. Dang, *et al.*, "An empirical investigation of missing data handling in cloud node failure prediction," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 1453–1464, 2022.

[2] J. Chen, X. He, Q. Lin, Y. Xu, H. Zhang, D. Hao, F. Gao, Z. Xu, Y. Dang, and D. Zhang, "An empirical investigation of incident triage for online service systems," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 111–120, IEEE, 2019.

[3] J. Soldani and A. Brogi, "Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey," *ACM Comput. Surv.*, vol. 55, Feb. 2022.

[4] S. Zhang, S. Xia, W. Fan, B. Shi, X. Xiong, Z. Zhong, M. Ma, Y. Sun, and D. Pei, "Failure diagnosis in microservice systems: A comprehensive survey and analysis," *ACM Transactions on Software Engineering and Methodology*, vol. 35, no. 1, pp. 1–55, 2025.

[5] L. Pham, H. Ha, and H. Zhang, "Baro: Robust root cause analysis for microservices via multivariate bayesian online change point detection," *Proceedings of the ACM on Software Engineering*, vol. 1, no. FSE, pp. 2214–2237, 2024.

[6] C. Zhang, Z. Dong, X. Peng, B. Zhang, and M. Chen, "Trace-based multi-dimensional root cause localization of performance issues in microservice systems," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, (New York, NY, USA), Association for Computing Machinery, 2024.

[7] C.-M. Lin, C. Chang, W.-Y. Wang, K.-D. Wang, and W.-C. Peng, "Root cause analysis in microservice using neural granger causal discovery," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 206–213, 2024.

[8] Y. Wang, Z. Zhu, Q. Fu, Y. Ma, and P. He, "Mrca: Metric-level root cause analysis for microservices via multi-modal data," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pp. 1057–1068, 2024.

[9] L. Zheng, Z. Chen, J. He, and H. Chen, "Mulan: multi-modal causal structure learning and root cause analysis for microservice systems," in *Proceedings of the ACM Web Conference 2024*, pp. 4107–4116, 2024.

[10] S. Zhang, P. Jin, Z. Lin, Y. Sun, B. Zhang, S. Xia, Z. Li, Z. Zhong, M. Ma, W. Jin, D. Zhang, Z. Zhu, and D. Pei, "Robust failure diagnosis of microservice system through multimodal data," *IEEE Transactions on Services Computing*, vol. 16, no. 6, pp. 3851–3864, 2023.

[11] G. Somashekar, A. Dutt, M. Adak, T. Lorido Botran, and A. Gandhi, "Gamma: Graph neural network-based multi-bottleneck localization for microservices applications," in *Proceedings of the ACM Web Conference 2024*, pp. 3085–3095, 2024.

[12] R. Xin, P. Chen, and Z. Zhao, "Causalrca: Causal inference based precise fine-grained root cause localization for microservice applications," *Journal of Systems and Software*, vol. 203, p. 111724, 2023.

[13] Y. Jiang, C. Zhang, S. He, Z. Yang, M. Ma, S. Qin, Y. Kang, Y. Dang, S. Rajmohan, Q. Lin, *et al.*, "Xpert: Empowering incident management with query recommendations via large language models," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pp. 1–13, 2024.

[14] D. Roy, X. Zhang, R. Bhave, C. Bansal, P. Las-Casas, R. Fonseca, and S. Rajmohan, "Exploring llm-based agents for root cause analysis," in *Companion proceedings of the 32nd ACM international conference on the foundations of software engineering*, pp. 208–219, 2024.

[15] C. Wang, X. Zhang, R. Lu, X. Lin, X. Zeng, X. Zhang, Z. An, G. Wu, J. Gao, C. Tian, *et al.*, "Towards llm-based failure localization in production-scale networks," in *Proceedings of the ACM SIGCOMM 2025 Conference*, pp. 496–511, 2025.

[16] Z. Yu, M. Ma, C. Zhang, S. Qin, Y. Kang, C. Bansal, S. Rajmohan, Y. Dang, C. Pei, D. Pei, *et al.*, "Monitorassistant: Simplifying cloud service monitoring via large language models," in *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, pp. 38–49, 2024.

[17] T. Ahmed, S. Ghosh, C. Bansal, T. Zimmermann, X. Zhang, and S. Rajmohan, "Recommending root-cause and mitigation steps for cloud incidents using large language models," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pp. 1737–1749, IEEE, 2023.

[18] L. Zhang, Y. Zhai, T. Jia, C. Duan, S. Yu, J. Gao, B. Ding, Z. Wu, and Y. Li, "Thinkfl: Self-refining failure localization for microservice systems via reinforcement fine-tuning," *ACM Transactions on Software Engineering and Methodology*, 2025.

[19] S. Shajarian, S. Khorsandroo, and M. Abdelsalam, "Poster: Intelligent network management: Rag-enhanced llms for log analysis, troubleshooting, and documentation," in *Proceedings of the 20th International Conference on emerging Networking EXperiments and Technologies*, pp. 27–28, 2024.

[20] Y. Li, Y. Wu, J. Liu, Z. Jiang, Z. Chen, G. Yu, and M. R. Lyu, "Coca: Generative root cause analysis for distributed systems with code knowledge," in *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pp. 1346–1358, IEEE, 2025.

[21] X. Zhou, G. Li, Z. Sun, Z. Liu, W. Chen, J. Wu, J. Liu, R. Feng, and G. Zeng, "D-bot: Database diagnosis system using large language models," *Proceedings of the VLDB Endowment*, vol. 17, no. 10, pp. 2514–2527, 2024.

[22] C. Pei, Z. Wang, F. Liu, Z. Li, Y. Liu, X. He, R. Kang, T. Zhang, J. Chen, J. Li, *et al.*, "Flow-of-action: Sop enhanced llm-based multi-agent system for root cause analysis," in *Companion Proceedings of the ACM on Web Conference 2025*, pp. 422–431, 2025.

[23] W. Zhang, H. Guo, J. Yang, Z. Tian, Y. Zhang, Y. Chaoran, Z. Li, T. Li, X. Shi, L. Zheng, *et al.*, "mabc: multi-agent blockchain-inspired collaboration for root cause analysis in micro-services architecture," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4017–4033, 2024.

[24] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24824–24837, 2022.

[25] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2022.

[26] Y. Xie, K. Kawaguchi, Y. Zhao, J. X. Zhao, M.-Y. Kan, J. He, and M. Xie, "Self-evaluation guided beam search for reasoning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 41618–41650, 2023.

[27] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," *Advances in neural information processing systems*, vol. 36, pp. 11809–11822, 2023.

[28] Z. Wang, Z. Liu, Y. Zhang, A. Zhong, J. Wang, F. Yin, L. Fan, L. Wu, and Q. Wen, "Rcagent: Cloud root cause analysis by autonomous agents with tool-augmented large language models," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 4966–4974, 2024.

[29] C. Qian, X. Cong, C. Yang, W. Chen, Y. Su, J. Xu, Z. Liu, and M. Sun, "Communicative agents for software development," *CoRR*, vol. abs/2307.07924, 2023.

[30] T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R. Wang, Y. Yang, S. Shi, and Z. Tu, "Encouraging divergent thinking in large language models through multi-agent debate," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, eds.), (Miami, Florida, USA), pp. 17889–17904, Association for Computational Linguistics, Nov. 2024.

[31] Z. Liu, Y. Zhang, P. Li, Y. Liu, and D. Yang, "Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization," *CoRR*, vol. abs/2310.02170, 2023.

[32] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *The eleventh international conference on learning representations*, 2022.

[33] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[34] L. McInnes, J. Healy, S. Astels, *et al.*, "hdbscan: Hierarchical density based clustering.," *J. Open Source Softw.*, vol. 2, no. 11, p. 205, 2017.

[35] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[36] L. Pham, H. Zhang, H. Ha, F. Salim, and X. Zhang, "Rcaeval: A benchmark for root cause analysis of microservice systems with telemetry data," in *Companion Proceedings of the ACM on Web Conference 2025*, pp. 777–780, 2025.