香港中文大學
The Chinese University of Hong Kong

# Reliability Engineering Challenges in Networking for AI

Hong Xu

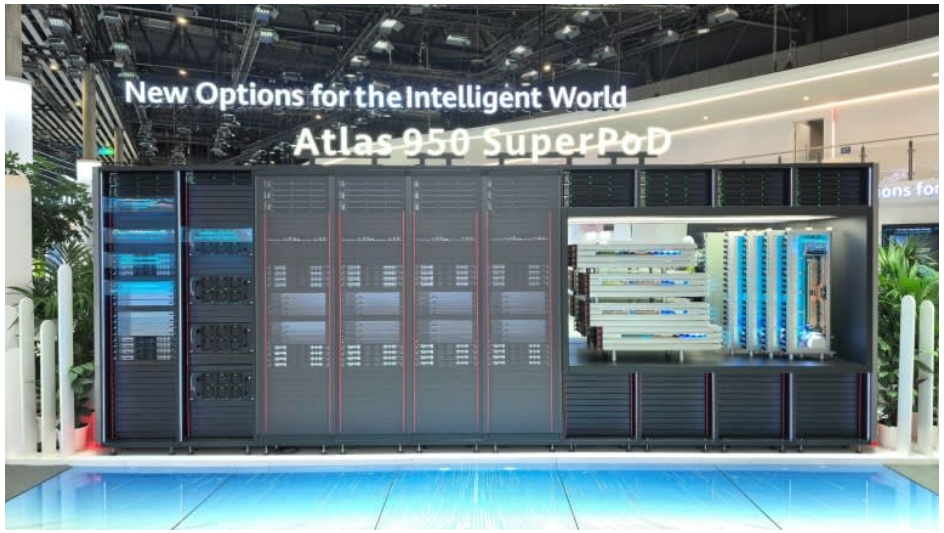
The Chinese University of Hong Kong

IRTFOPEN meeting, IETF-125

*March 17, 2026*

# Networking Infrastructures for AI

- Server-level: Interconnects
  - UCIe, CXL, NVLink-C2C, UB
- Rack-level: Interconnects
  - Unified memory: NVL72/144, Atlas 950 SuperPoD
- Cluster-level: Fabric
  - Optics integration
- Scale: O(100k) GPU training




Source: Nikkei

arXiv > cs > arXiv:2602.00277

Computer Science > Distributed, Parallel, and Cluster Computing

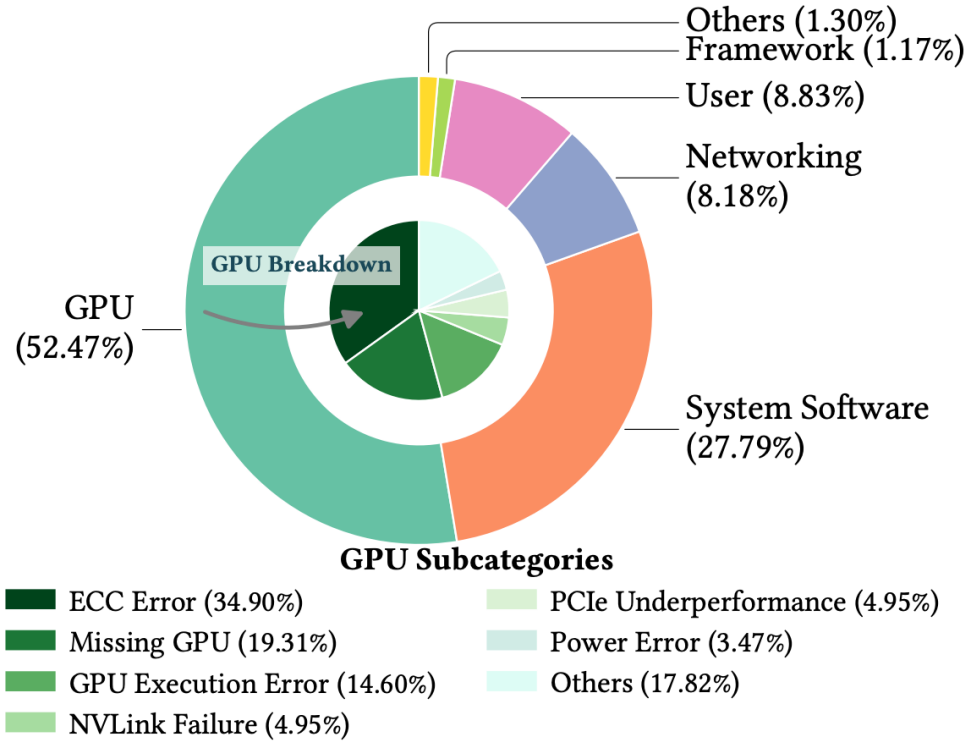
*[Submitted on 30 Jan 2026]*

**Training LLMs with Fault Tolerant HSDP on 100,000 GPUs**

# Challenge: Reliability Engineering

- Scale and complexity → Many faults across the hw-sw stack
- Distributed training/inference → Huge financial cost of faults
- Current practice is manual & tedius

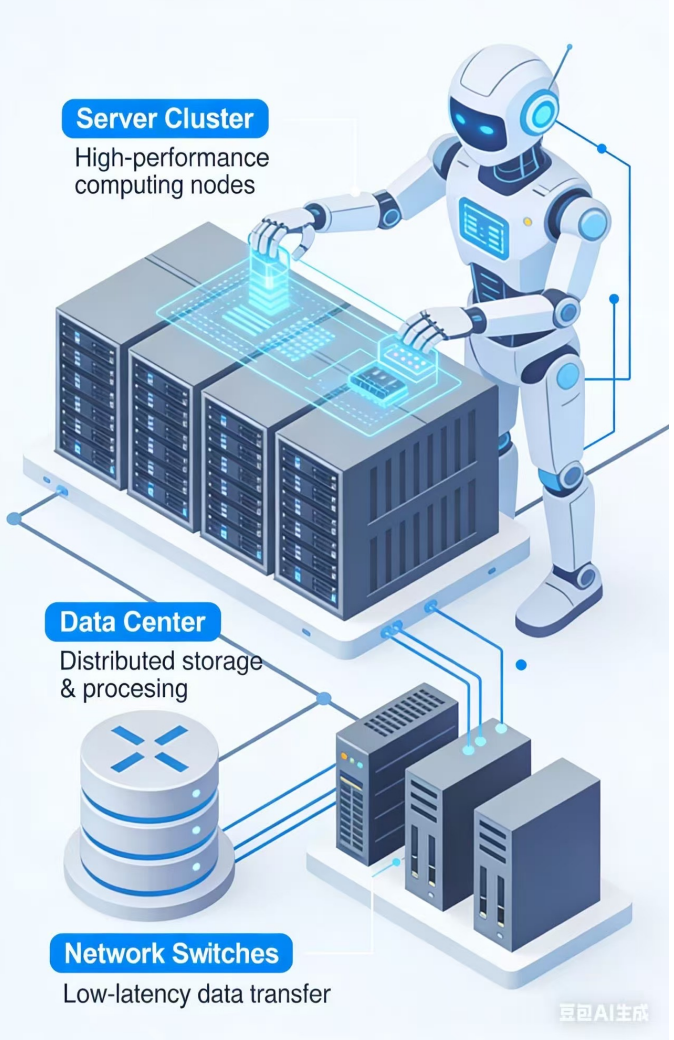
→ Can't keep up with rapidly-evolving AI infra




~1,300 real-world incident data from 3 production GPU clusters serving customer AI workloads at Azure (2023-04 to 2024-03) Yang et al., FSE'26; arXiv:2506.01481

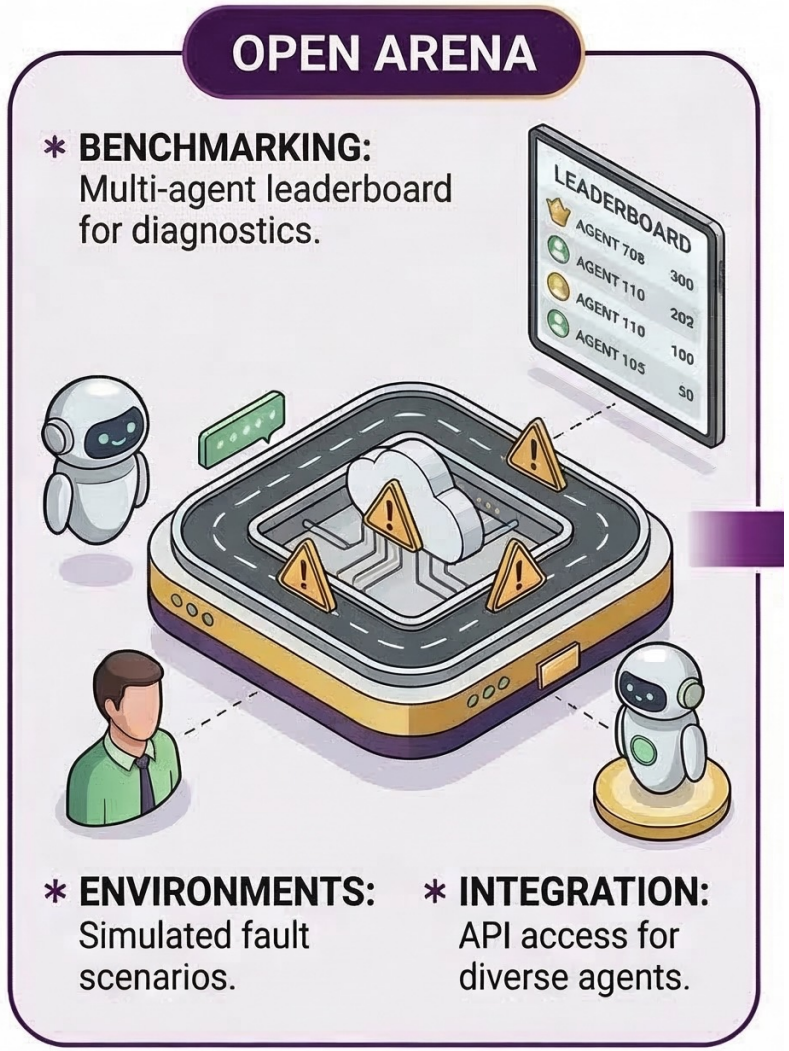
# Vision: Autonomous Agentic Troubleshooting for Infra

- Build an OpenClaw for NetOps/InfraOps
- *Q1: Eval: how to measure the agent?*
- *Q2: Design Paradigm: how to design the agent?*
- *Q3: System: how to run this agent?*



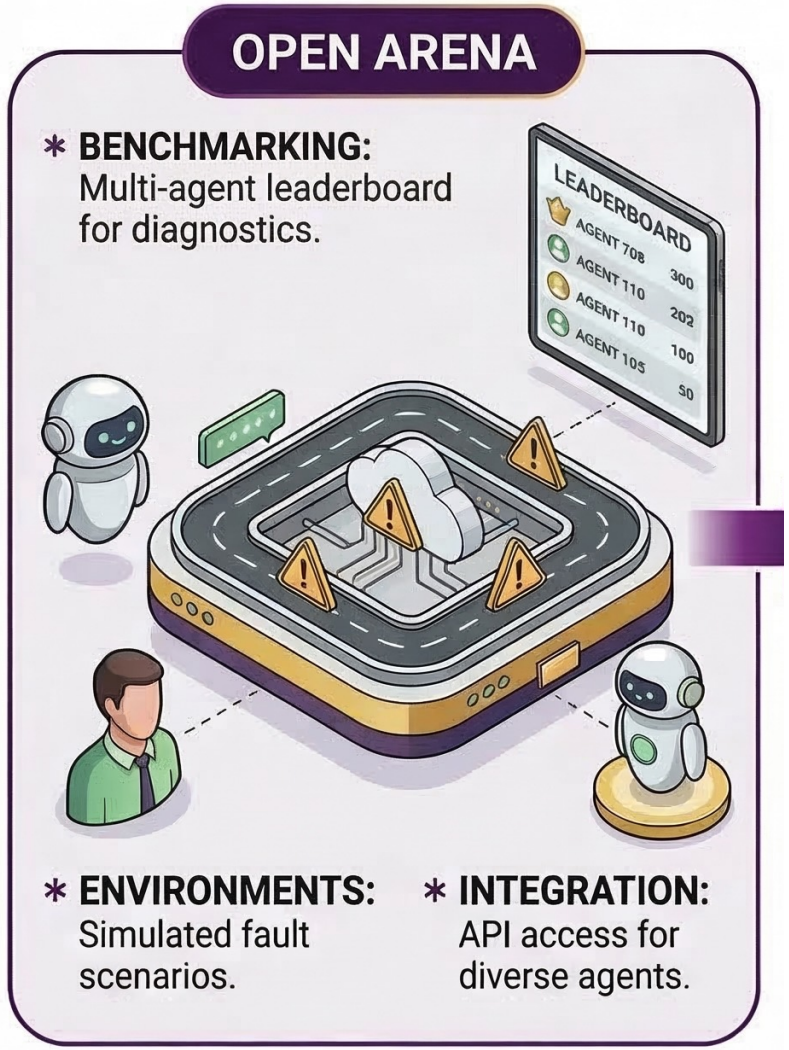

Created by Seedream 4.0

# SYSTEMATIC FRAMEWORK FOR AGENT-DRIVEN DIAGNOSTICS




Created by NanoBanana

# SYSTEMATIC FRAMEWORK FOR AGENT-DRIVEN DIAGNOSTICS



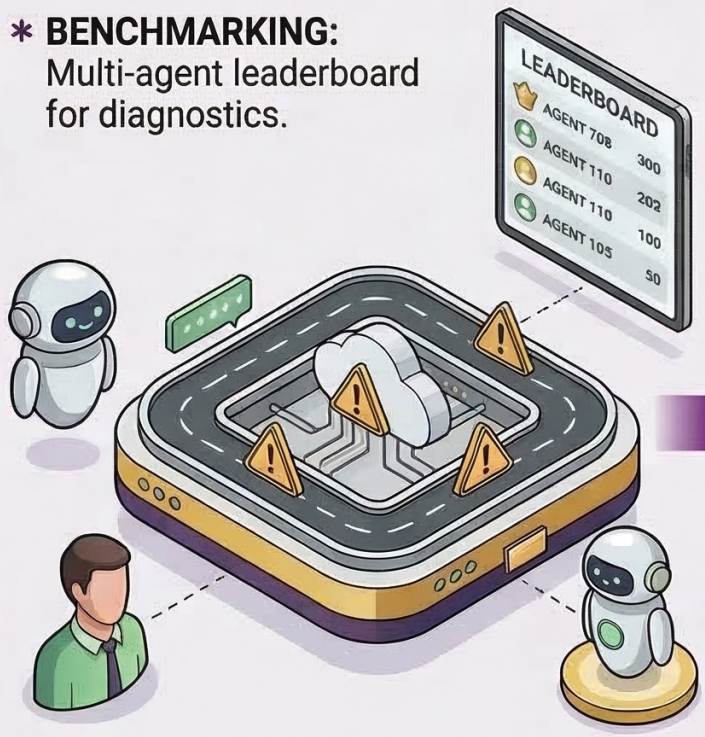

Created by NanoBanana

- Standardized reproducible benchmarking
- C1: Realistic comprehensive fault datasets
- C2: Faithful, sandboxed, interactive environment
  - Agents interact with the env
  - Emulation, simulation, testbed
- C3: Integration with production tooling
  - Pingmesh, Mycroft, etc.

# SYSTEMATIC FRAMEWORK FOR AGENT-DRIVEN DIAGNOSTICS

## OPEN ARENA

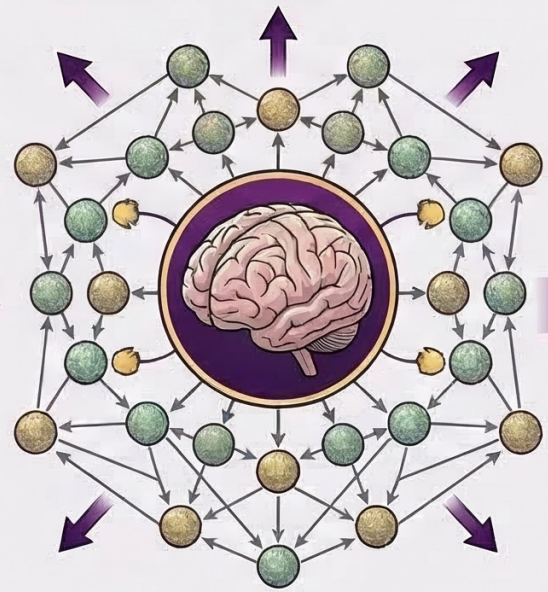
* **BENCHMARKING:** Multi-agent leaderboard for diagnostics.




* **ENVIRONMENTS:** Simulated fault scenarios.
* **INTEGRATION:** API access for diverse agents.

## AGENTIC PARADIGM

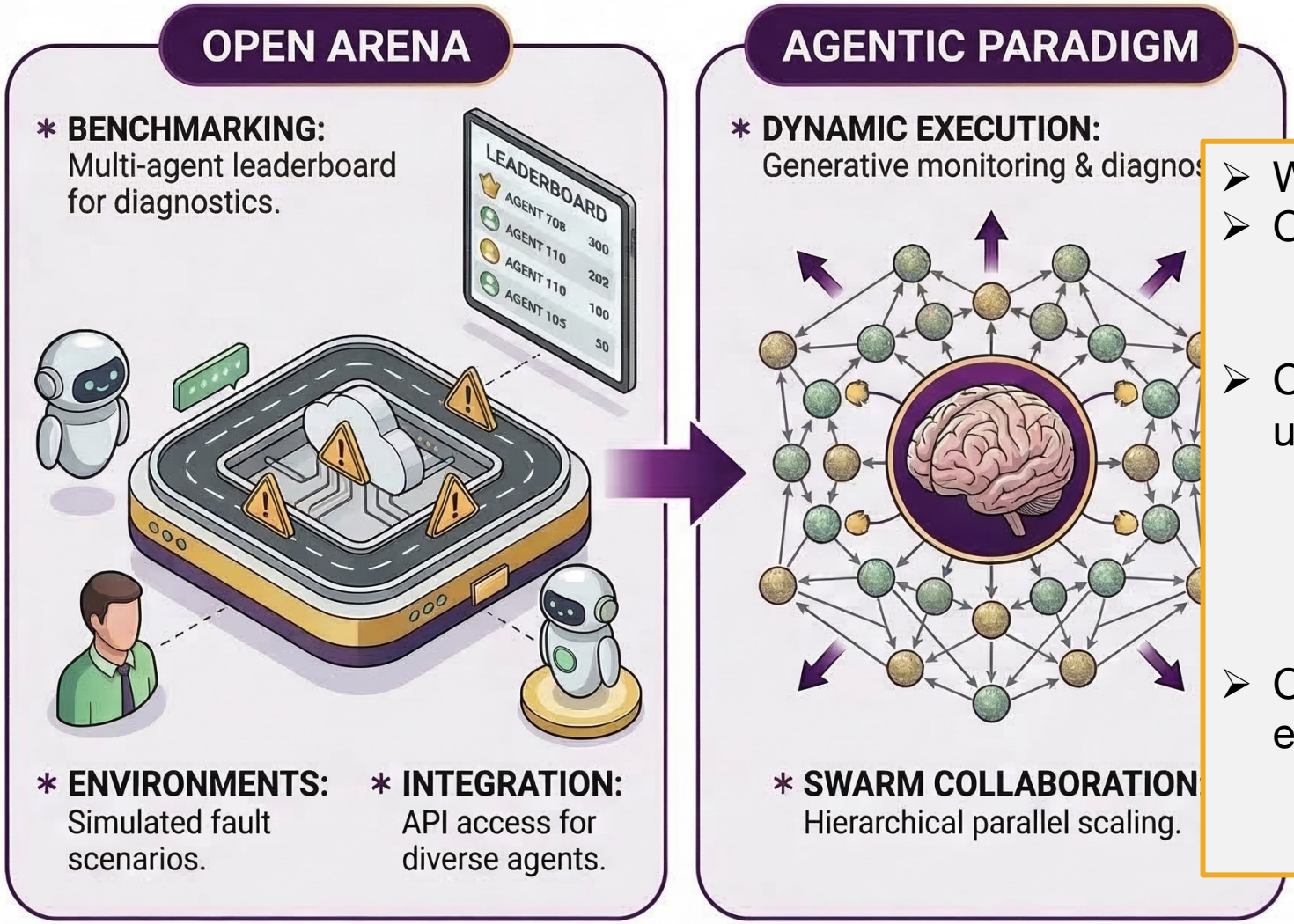
* **DYNAMIC EXECUTION:** Generative monitoring & diagnosis.



* **SWARM COLLABORATION:** Hierarchical parallel scaling.

Created by NanoBanana

# SYSTEMATIC FRAMEWORK FOR AGENT-DRIVEN DIAGNOSTICS



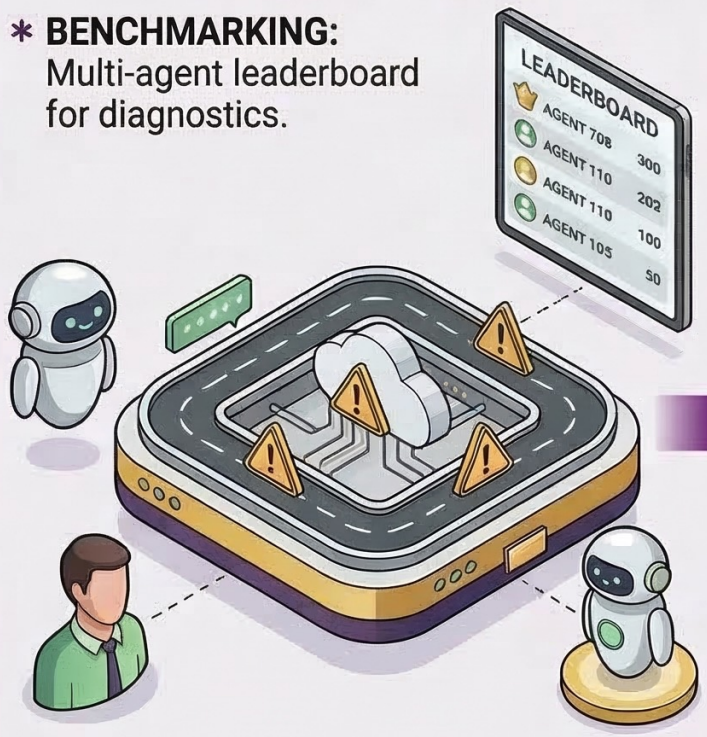

- Workflow Engineering
- C1: Context length bottleneck
  - Massive telemetry data from logs and metrics
- C2: Adaptive reasoning with proper tool use
  - Narrow down the scope, verify hypotheses, etc.
  - Sequential or parallel (check different pods of the network)
- C3: Distilling new knowledge from experience
  - Summarize new SOPs for unseen incidents

Created by NanoBanana

# SYSTEMATIC FRAMEWORK FOR AGENT-DRIVEN DIAGNOSTICS

## OPEN ARENA

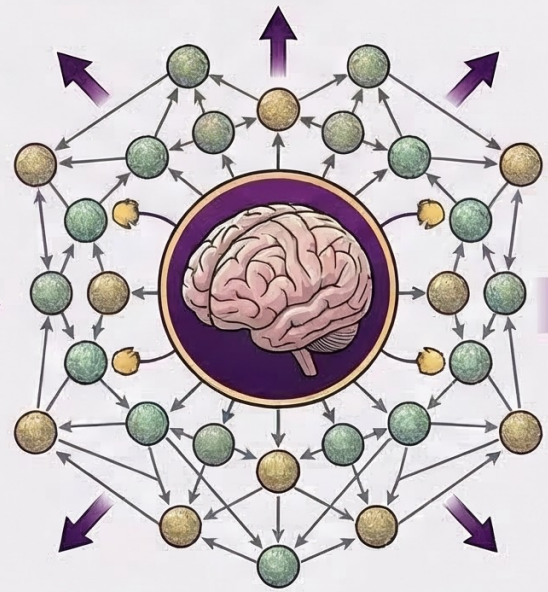
* **BENCHMARKING:** Multi-agent leaderboard for diagnostics.




* **ENVIRONMENTS:** Simulated fault scenarios.
* **INTEGRATION:** API access for diverse agents.

## AGENTIC PARADIGM

* **DYNAMIC EXECUTION:** Generative monitoring & diagnosis.



* **SWARM COLLABORATION:** Hierarchical parallel scaling.

## SERVING & ASSURANCE

* **SYSTEM PERFORMANCE OPTIMIZATION:** Speculative decoding & cross-model KV-Cache.

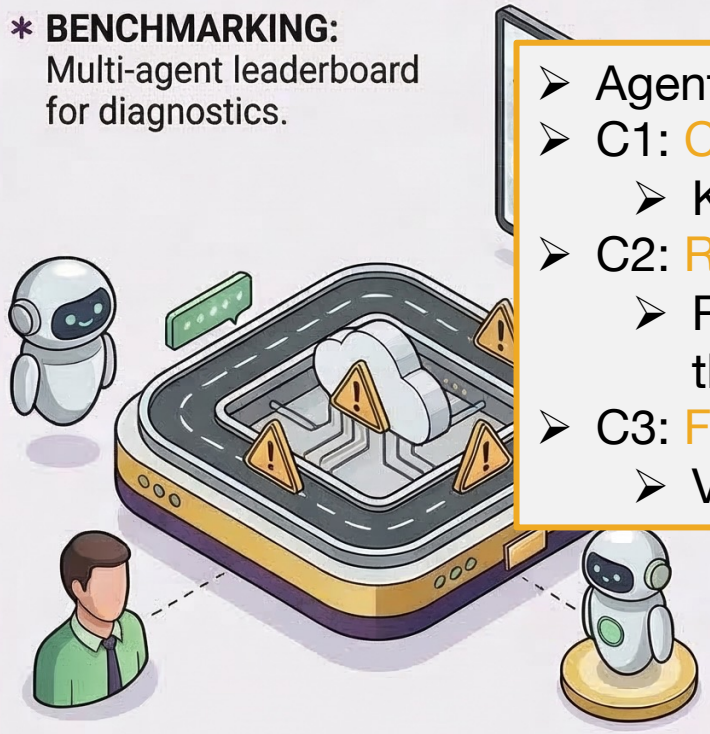

* **EFFICIENCY:** Resource Allocation & Throughput Maximization.
* **RELIABILITY:** Agentic red-teaming & Fault Tolerance.

Created by NanoBanana

# SYSTEMATIC FRAMEWORK FOR AGENT-DRIVEN DIAGNOSTICS

## OPEN ARENA

* **BENCHMARKING:** Multi-agent leaderboard for diagnostics.

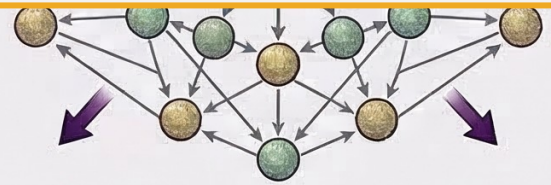


* **ENVIRONMENTS:** Simulated fault scenarios.
* **INTEGRATION:** API access for diverse agents.

## AGENTIC PARADIGM

* **DYNAMIC EXECUTION:**

- Agentic Serving Systems
- C1: Cross-model communication
  - KV cache, or non-KV cache
- C2: Robustness, assurance
  - Red-teaming, understand how the agent fails
- C3: Full DevOps cycle support
  - Versioning, testing, rollback, etc.



* **SWARM COLLABORATION:** Hierarchical parallel scaling.

## SERVING & ASSURANCE

* **SYSTEM PERFORMANCE OPTIMIZATION:** Speculative decoding & cross-model KV-Cache.


* **EFFICIENCY:** Resource Allocation & Throughput Maximization.
* **RELIABILITY:** Agentic red-teaming & Fault Tolerance.

Created by NanoBanana

# Our Progress Towards this Vision

- NetOpsArena: ongoing
- TSGuard [FSE'26]: A user-centric troubleshooting agent for AI workloads in the cloud
- NetOpsAI: A troubleshooting agent for networks deployed and running at a leading AI company in China
- Mycroft [SOSP'25]: Tracing tool for collective communication deployed and running at ByteDance

# TSGuard: Automated User-Centric Incident Diagnosis for AI Workloads in the Cloud

Yitao Yang, Yangtao Deng, Yifan Xiong,
Baochun Li, HX, Peng Cheng

*FSE'26*

香港中文大學
The Chinese University of Hong Kong

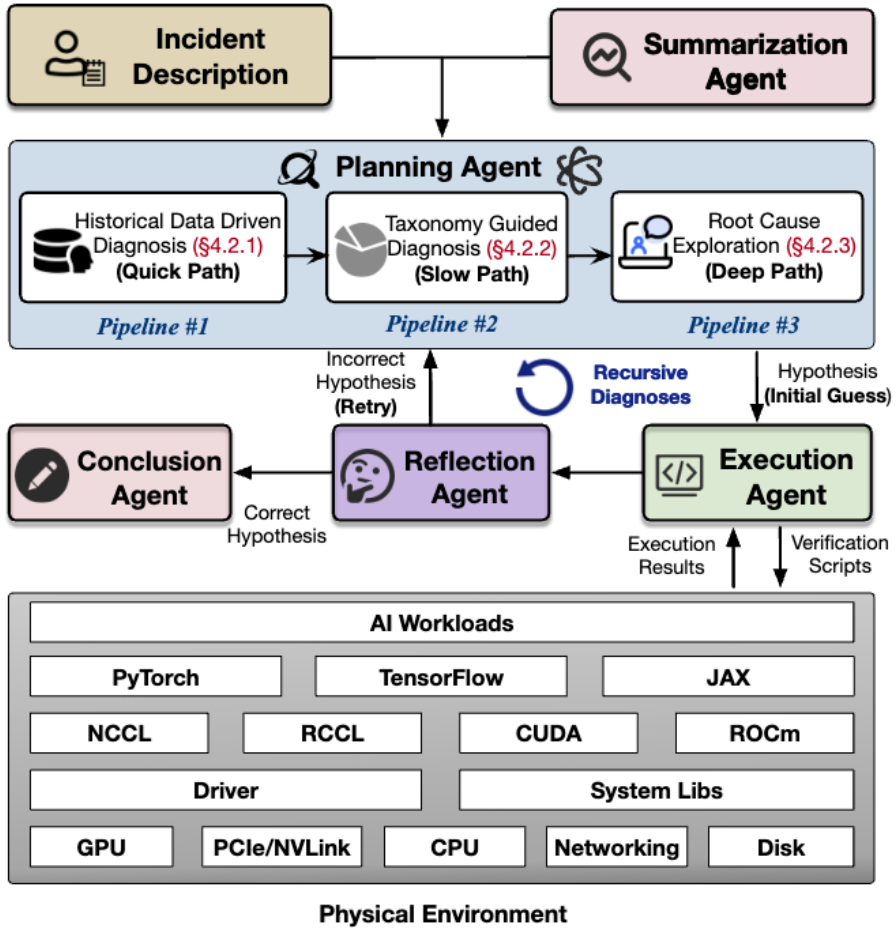
Microsoft

UNIVERSITY OF TORONTO

# TSGuard Architecture & Key Ideas

- User-Centric Incident Diagnosis
  - Empowers Users: Provides user-side diagnosis with immediate feedback.
  - Reduces On-Call Burden: Automatically intercepts and resolves recurring incidents, easing the workload for human support teams.

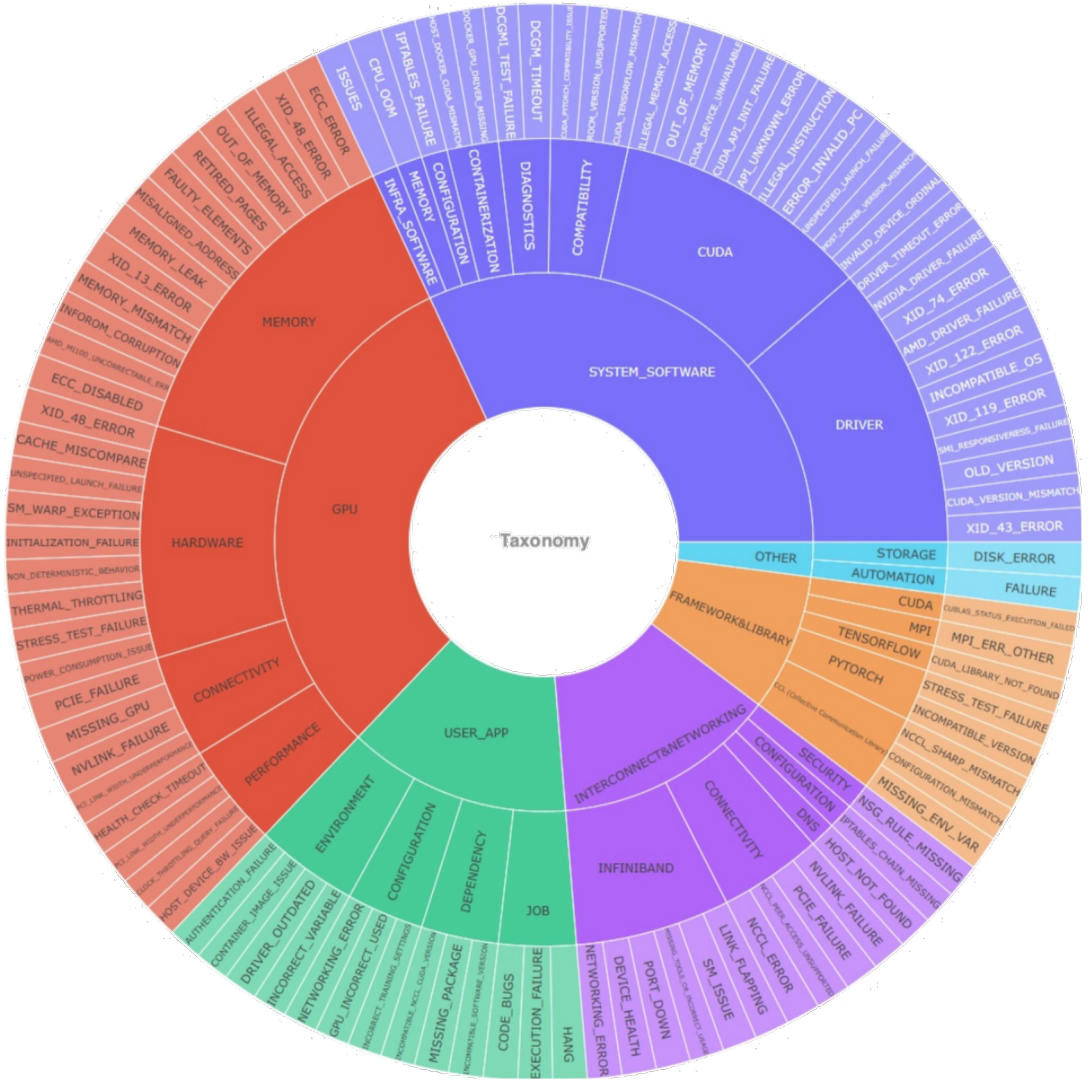
- System Workflow: A Two-Phase Approach
  - Phase 1 (Offline): Knowledge Consolidation (Building the Agent's "Brain").
  - Phase 2 (Online): Tiered Diagnostic Pipeline (Executing the Action).




Architectural Overview of TSGuard Online Phase

# Phase 1: Offline Knowledge Consolidation

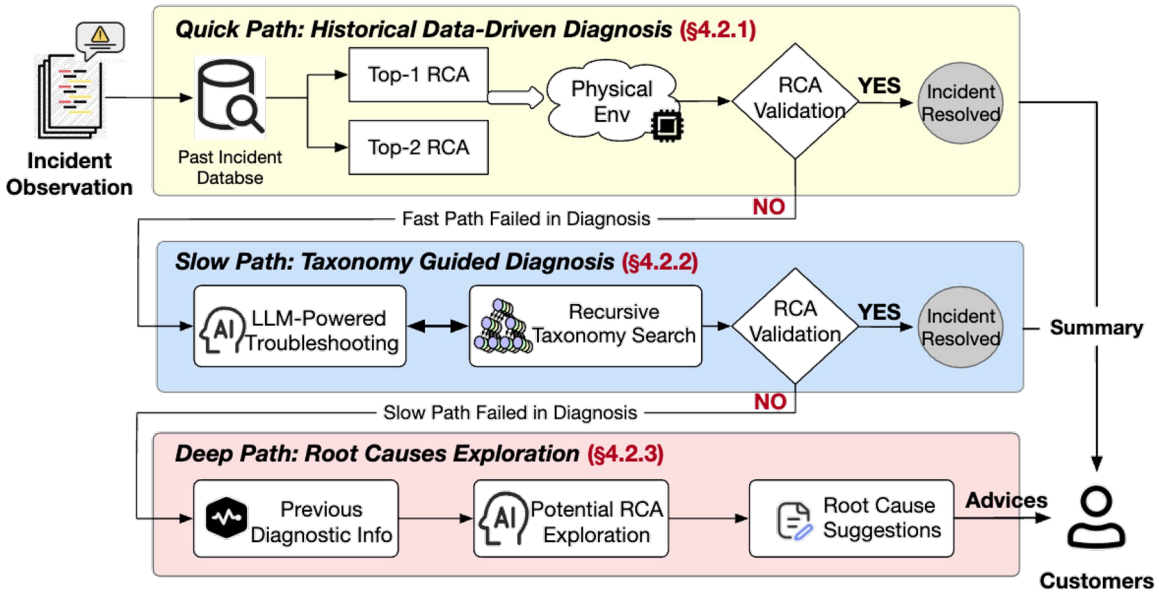
- **Historical Incident Database**
  - One-year production incidents (2023-2024) from three GPU clusters of Microsoft Azure.
- **Incident Taxonomy**
  - Transforms unstructured postmortems into structured, actionable reasoning paths for the planning agent.
- **Domain-Specific Rules**
  - Encapsulates on-call experience into text-based rules to guide diagnosis and prevent agent hallucinations.




Visualization of the Incident Taxonomy

# Phase 2: Online Tiered Diagnostic Pipeline

- Tier1: Historical Data-Driven Diagnosis
  - For Recurring Incidents: Fast-track resolution using RAG to match current observations with past identical failures.
- Tier2: Taxonomy Guided Diagnosis
  - For Complex Incidents: Step-by-step agentic reasoning systematically guided by the offline taxonomy (from Phase 1).
- Tier3: Root Causes Exploration
  - For Unseen Anomalies: Autonomous hypothesis generation and verification when predefined taxonomies are ineffective.

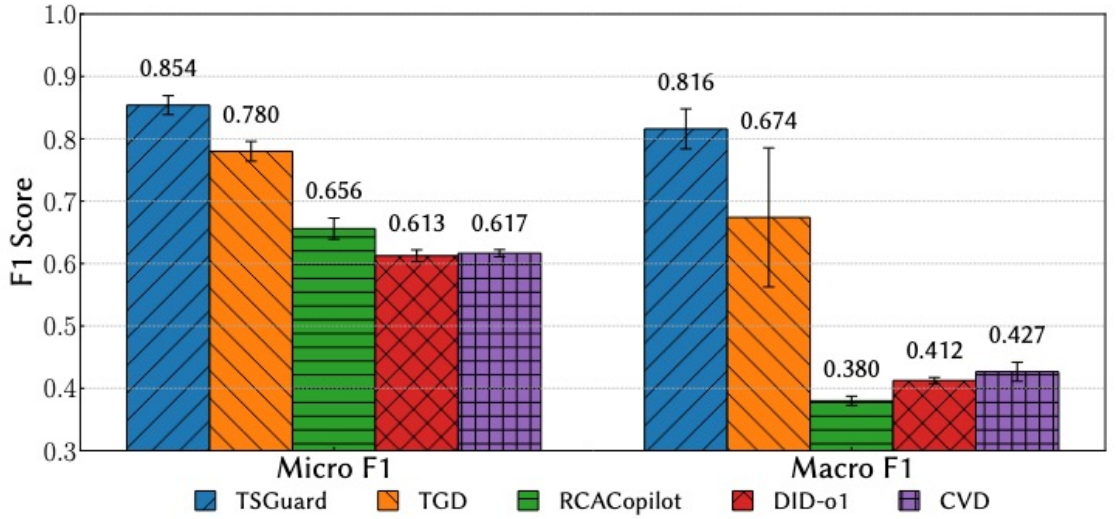


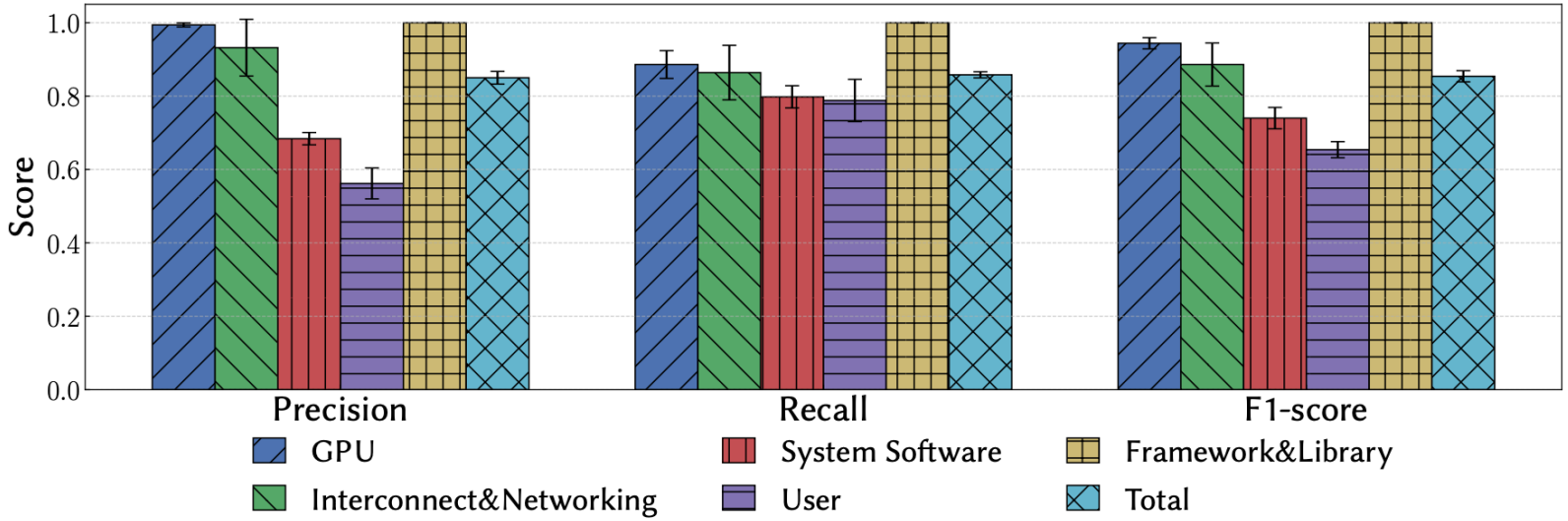

Online Diagnostic Procedures in TSGuard

# Evaluation with Real Incidents

TSGuard Achieves Highest Diagnostic Accuracy

Baselines:
RCACopilot (SOTA RAG), TGD (Tier2 only), DID-o1 (reasoning LLMs), CVD (Brute-force verification).



# NetOpsAI: AI-Driven NetOps System for A Leading AI Company in China

Yitao Yang

香港中文大學
The Chinese University of Hong Kong

# NetOpsAI: Running in A Leading AI Company in China

- Context-Aware Metadata Query
  - *Enumerate the names and network segments of all PoDs in [DC].*
- End-to-End Diagnosis/Root Cause Analysis
  - *Are there any recent BBONE/DCI link alerts within [Time Range]?*
- High-Fidelity Telemetry Analysis
  - *Query port link status in [time range] across all switches.*
- LLM-Empowered TSG Automation
  - *Batch diagnose network connectivity for [IP list].*

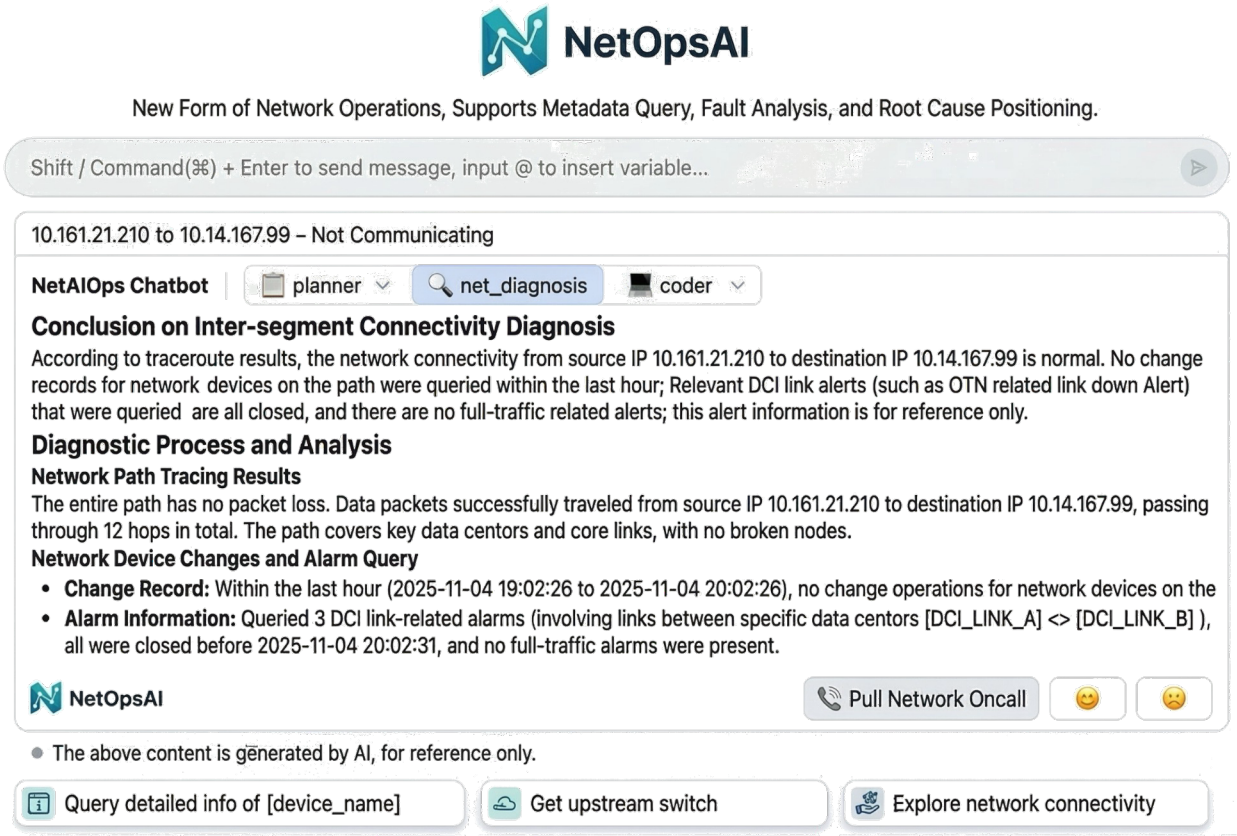



Illustration only; the real system differs

# Mycroft: Tracing Dependencies in Collective Communication Towards Reliable LLM Training

Yangtao Deng, Lei Zhang, Qinlong Wang, Xiaoyun Zhi, Xinlei Zhang, Zhuo Jiang, Haohan Xu, Lei Wang, Zuquan Song, Gaohong Liu, Yang Bai, Shuguang Wang, Wencong Xiao, Jianxi Ye, Minlan Yu, HX

*SOSP'25*

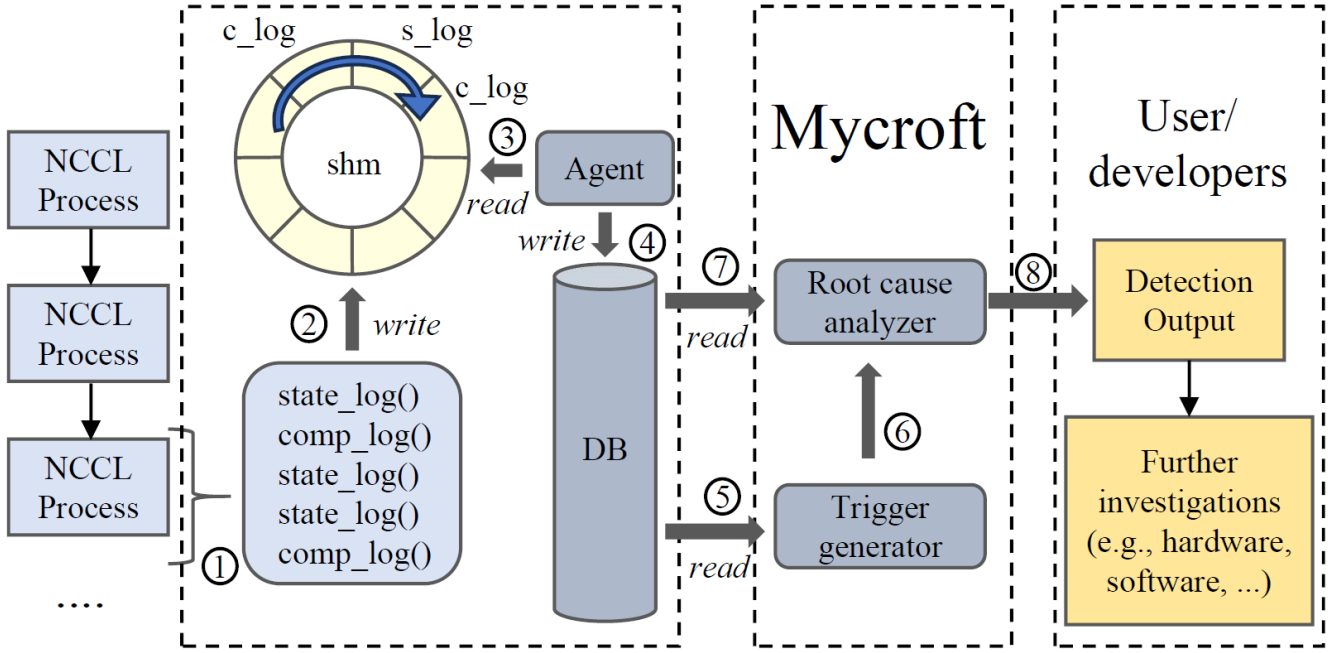
香港中文大學
The Chinese University of Hong Kong



HARVARD
UNIVERSITY

# Mycroft: Fine-grained Collective Level Tracing

- Coll-level tracing
  - *Real-time state logs:* Generated periodically when an op remains unfinished, to collect the aggregated Coll-level states
- Real-time anomaly detection
- Dependency-driven root cause analysis



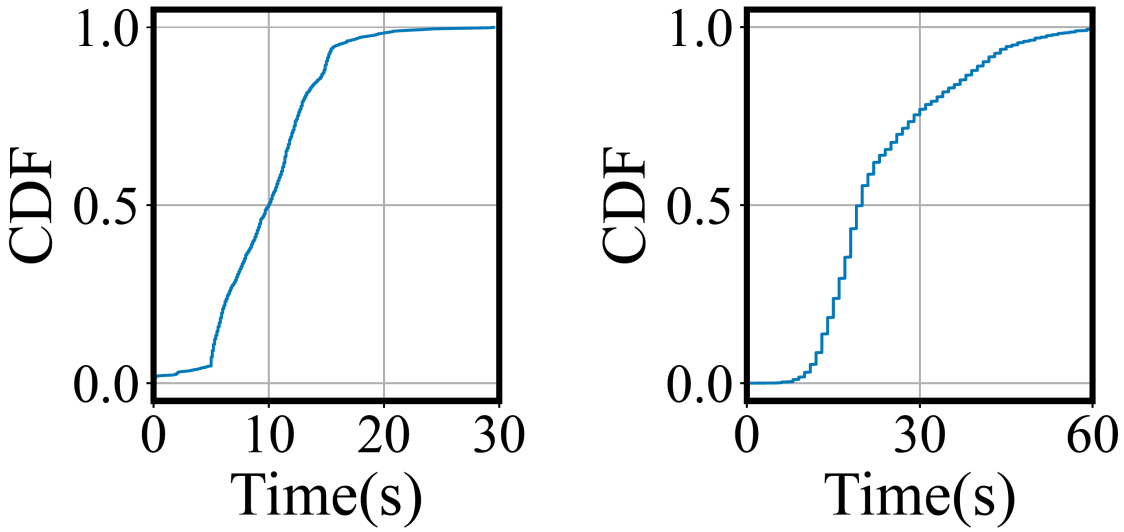
# Production Deployment

- Since 10/2024, running 24/7 for production training tasks at ByteDance
  - On a server with less than 64 CPU cores and 100GB memory
  - Monitoring tasks with 128+ GPUs
  - Up to 20 concurrent training tasks, generating 10TB tracing data per day
  - Occupying fixed 512MB space in shared memories in each host's containers

- Integration with other debugging systems
  - py-spy: A call stack sampling profiler for Python programs
  - Flight Recorder: Traces of the latest $N$ ops in a in a ring buffer (op ID, input and output sizes, execution state, and communication process group ID)

# Real-Time Performance

- Responsiveness during training




Trigger time and root cause analysis time distributions in ByteDance production

- **1,253 root cause analyses during 11/2024 and 12/2024. Only one problematic network flow in 705 cases**
- **90% were detected within 15 seconds, and 60% analyses completed within 20 seconds**

# Conclusion

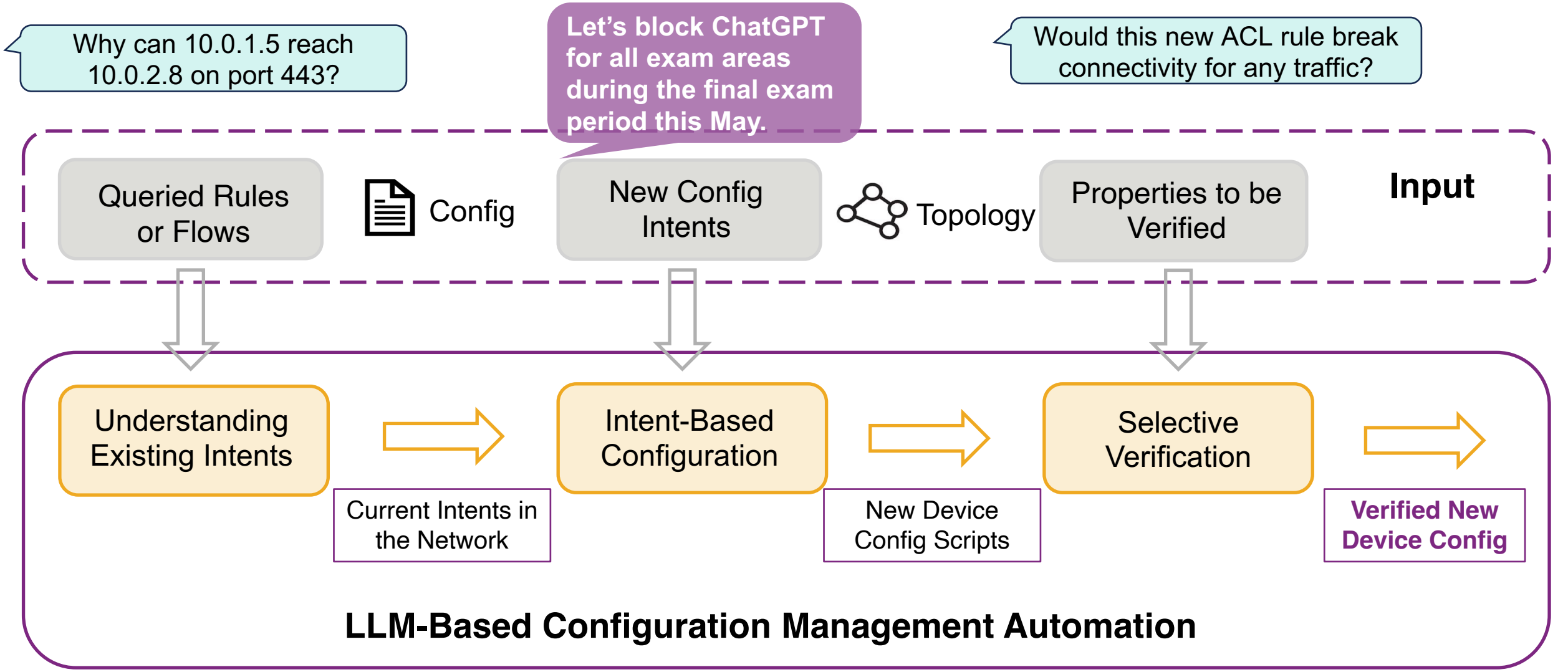
- A great time to work on reliability engineering for networking and AI infra
  - Collaboration, datasets, welcome!

- Look forward to making more impact
  - Open source NetOpsArena soon

TSGuard https://arxiv.org/abs/2506.01481

Mycroft https://dl.acm.org/doi/10.1145/3731569.3764848

# One More Thing…

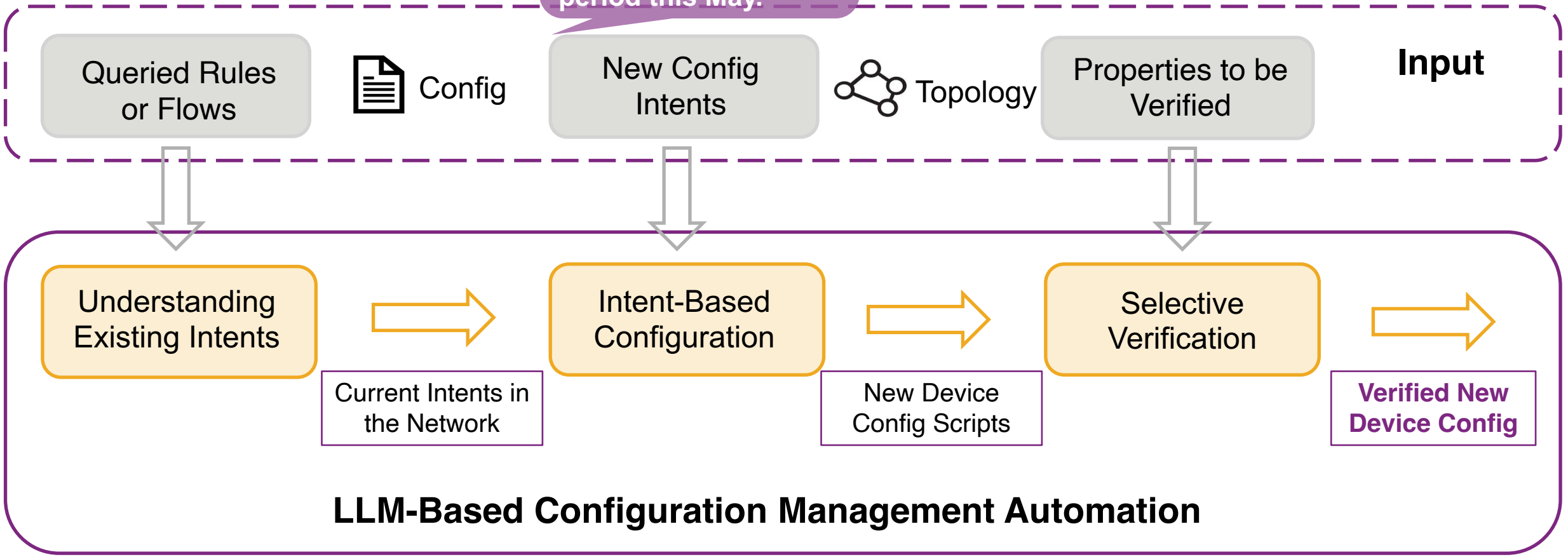


# One More Thing…

Check out my student's talk here: Wenlong Ding, https://datatracker.ietf.org/meeting/125/materials/agenda-125-nmrg

Why can 10.0.1.5 10.0.2.8 on port 4

during the final exam period this May.




THANK YOU!