sigcomm 2025

Towards LLM-Based Failure Localization in Production-Scale Networks

Chenxu Wang, Xumiao Zhang, Runwei Lu, Xianshang Lin, Xuan Zeng, Xinlei Zhang, Zhe An, Gongwei Wu, Jiaqi Gao, Chen Tian, Guihai Chen, Guyue Liu, Yuhong Liao, Tao Lin, Dennis Cai, Ennan Zhai

南京大学 NANJING UNIVERSITY

Alibaba Cloud

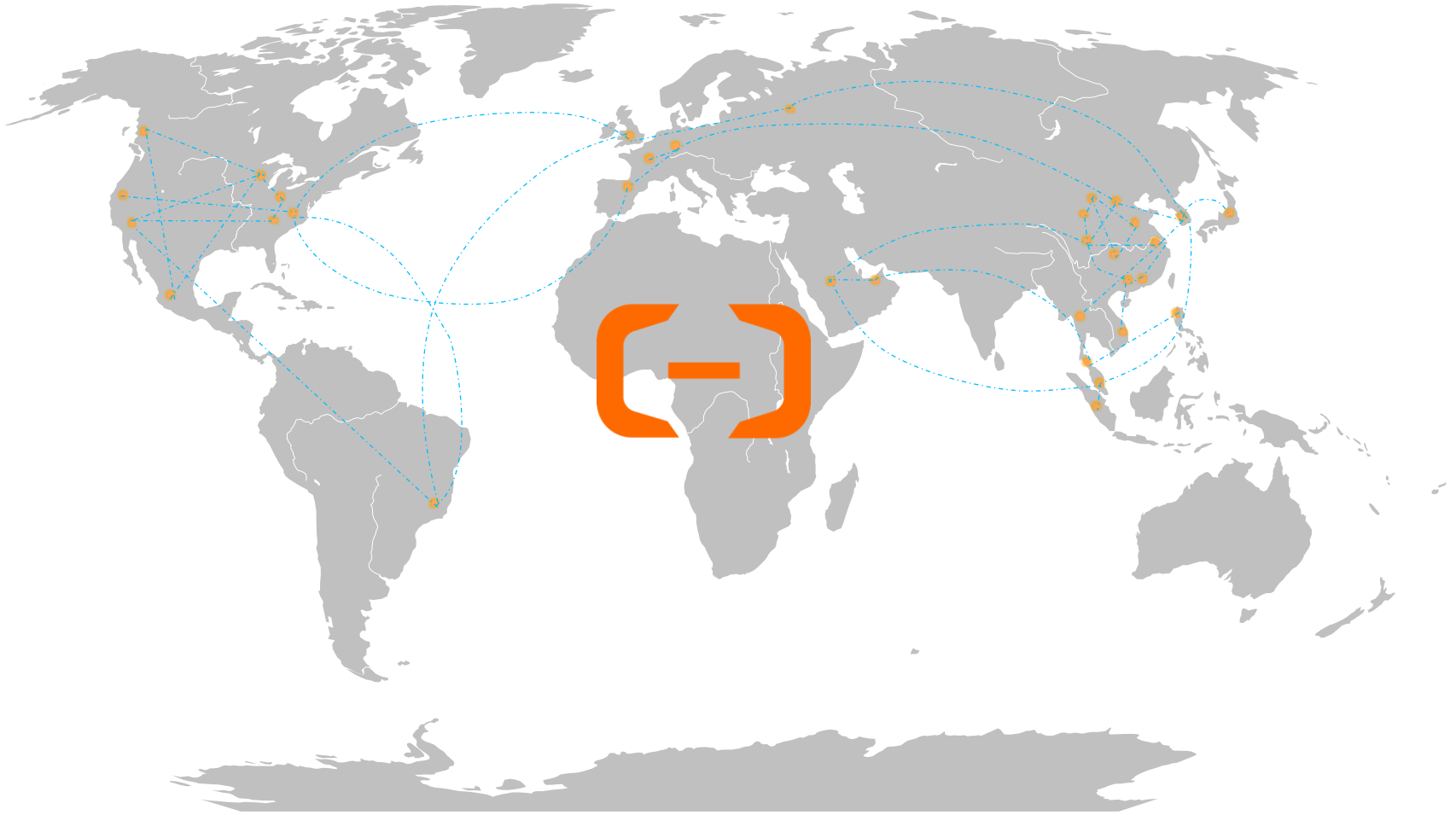
上海纽约大学 NYU SHANGHAI

北京大学 PEKING UNIVERSITY

Alibaba Cloud operates a global infrastructure

87 data centers

29 regions



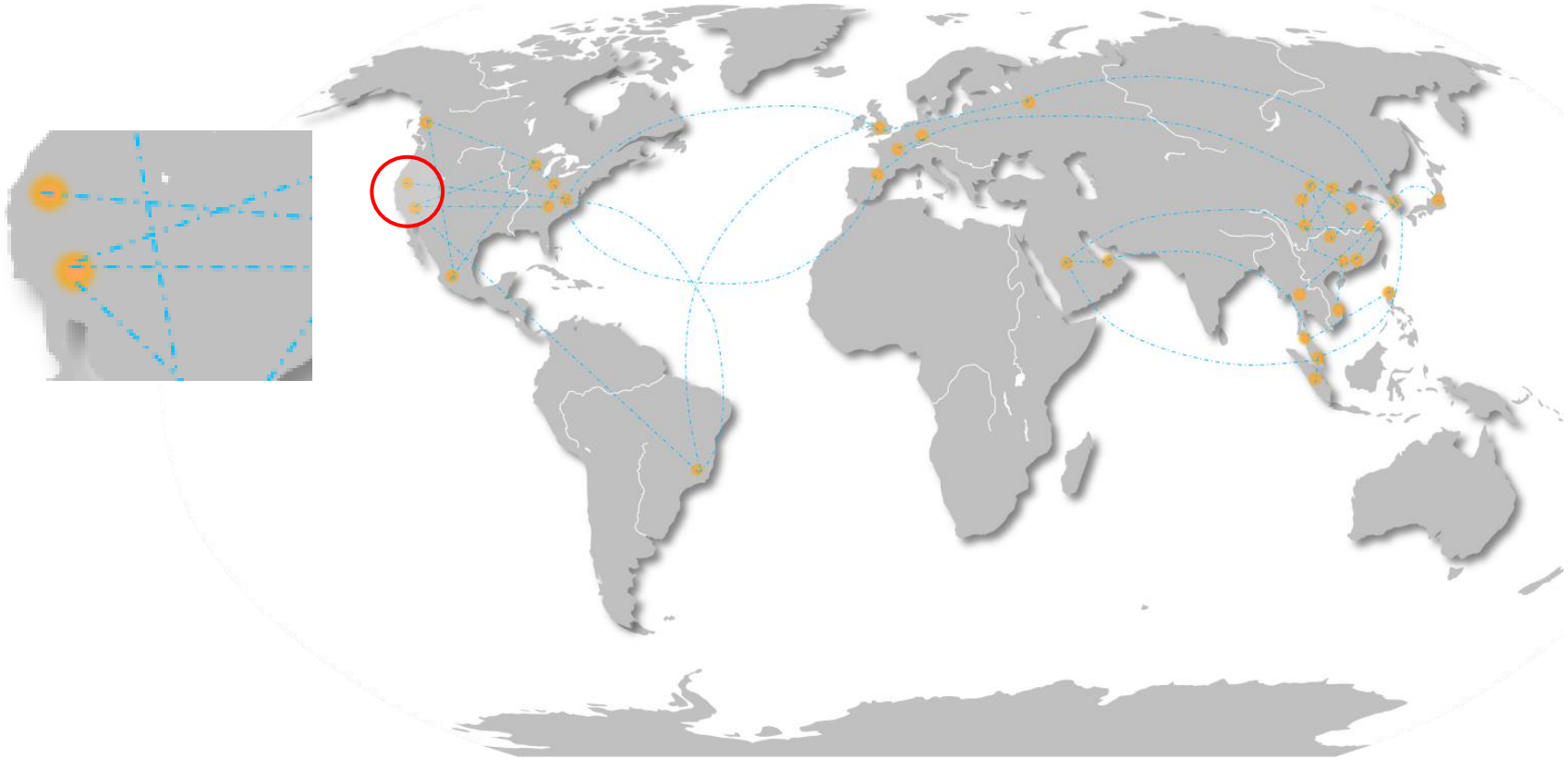
Alibaba Cloud operates a global infrastructure

Given such a scale, network incidents in **physical network** are inevitable.

~10^4 devices

~10^5 cards

~10^5 ports/links



Operators have been armed

For years, we have followed the state-of-the-art and developed various monitoring, failure localization and mitigation tools.

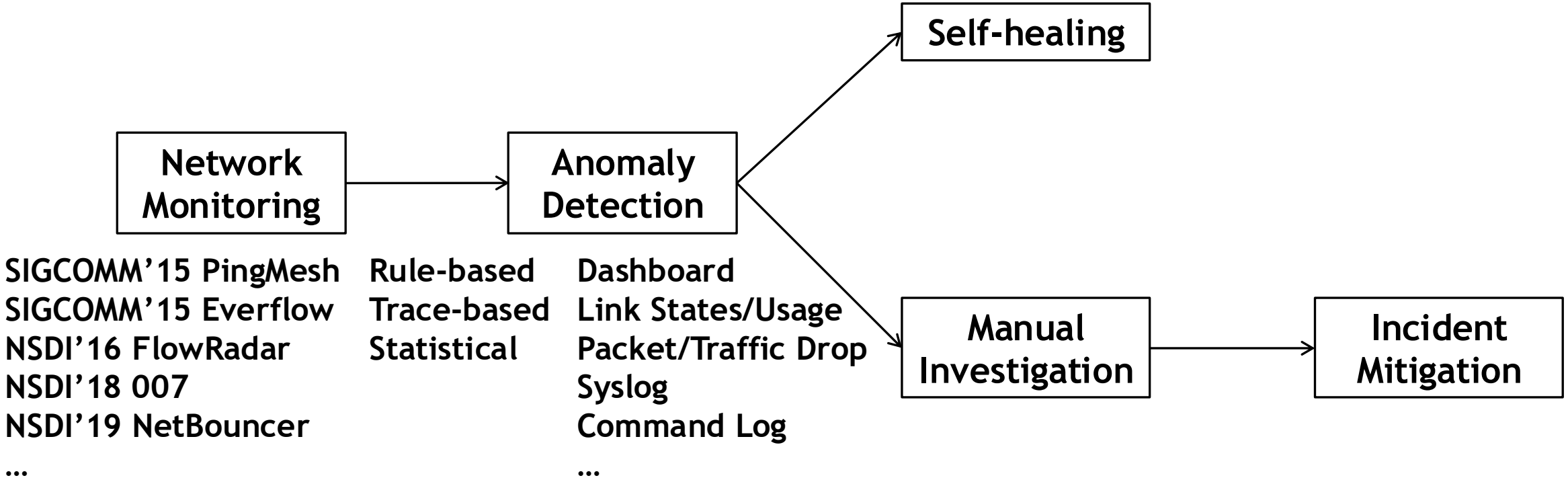
SIGCOMM'15 PingMesh
SIGCOMM'15 Everflow
NSDI'16 FlowRadar
NSDI'18 007
NSDI'19 NetBouncer
…

Rule-based
Trace-based
Statistical

Dashboard
Link States/Usage
Packet/Traffic Drop
Syslog
Command Log
…

Operators have been armed

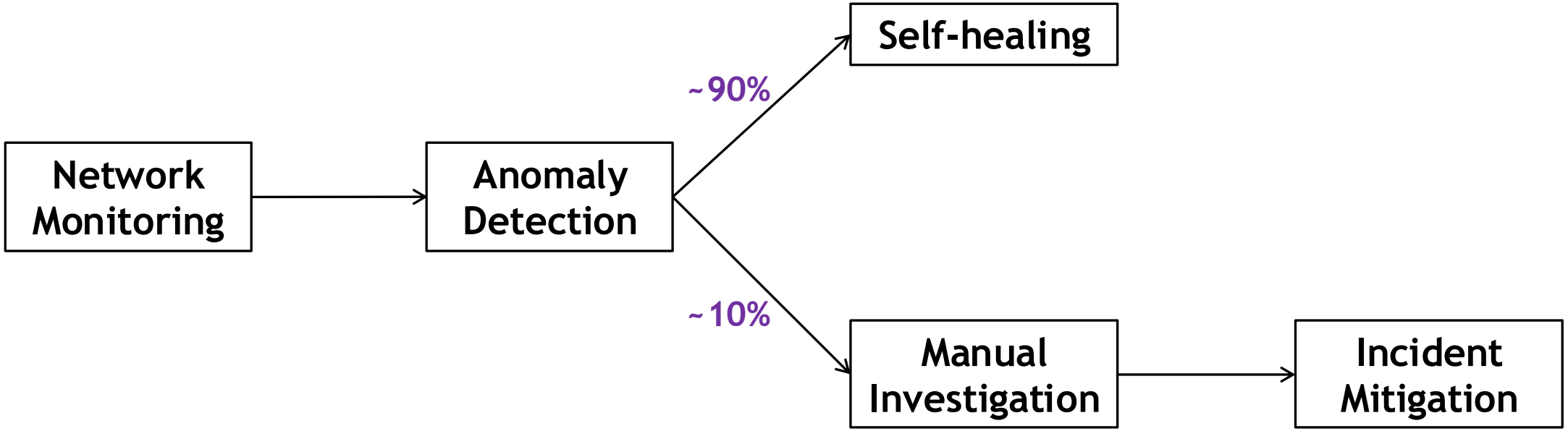
For years, we have followed the state-of-the-art and developed various monitoring, failure localization and mitigation tools.



And we integrate them into our incident management (IM) workflow.

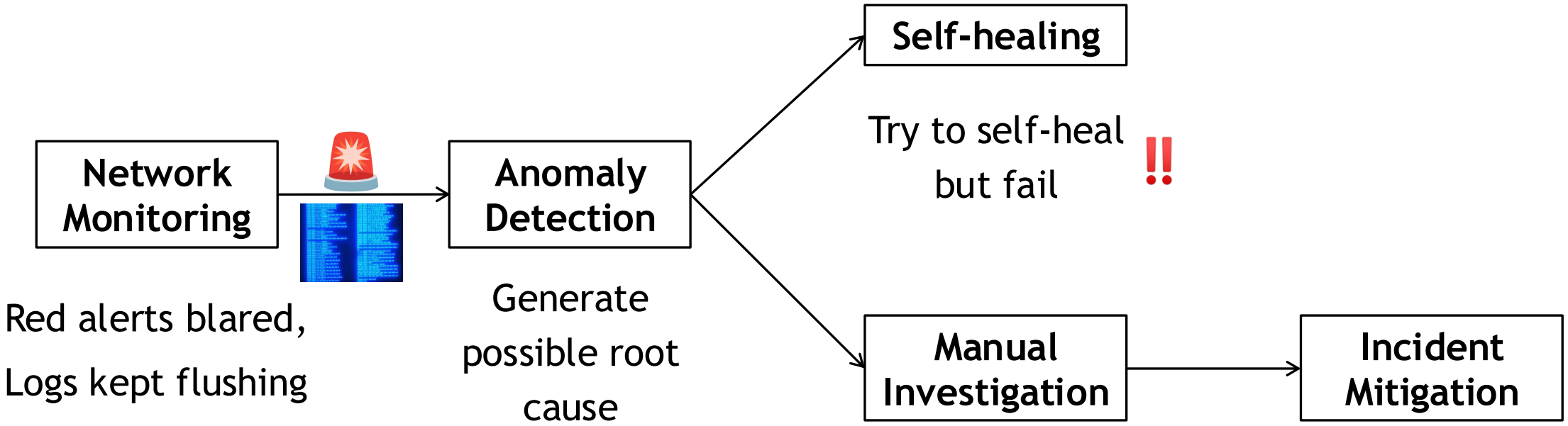
Operators have been armed, but

Self-healing is first attempted but some incidents need operators' attention.



Imagine such a case

After all predefined automated solutions were executed, the problem persisted.



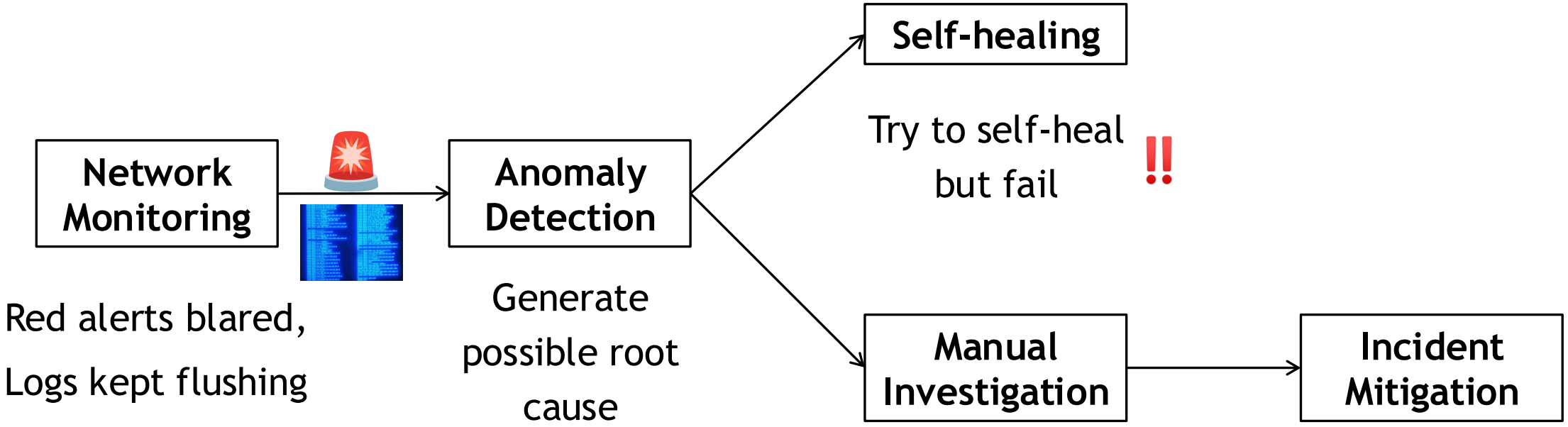
What else can we do to assist human operators, against the sea of data…?

How to help operators?

Basically, we have two options:

- *O1*: Build one more telemetry/algorithm/rule to monitor/locate and try to mitigate the incident automatically *instead of* operators.

  Sounds great, but once it failed, we enter the same problem.
- *O2*: Build one tool that try to assist the manual mitigation process.

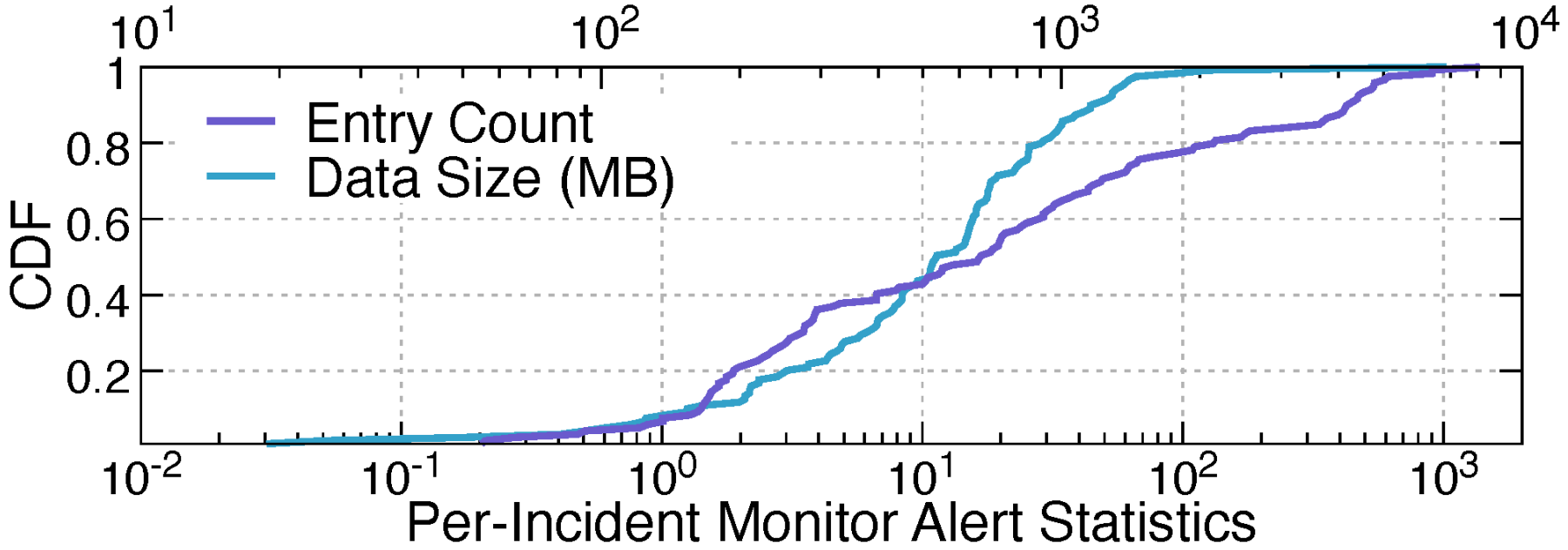


Our insights

1. We assist, not replace. Human operators are the final defense.

To help, to conquer the sea of data

Plan: To help operators understand what happened ASAP from the piled monitoring data.



That's where LLM came into our views.

LLM becomes a promising solution

LLMs address the inherent limitations of both traditional automation and human-led analysis.

1

Native Text Processing Capability

Natively understands and reasons with the human-readable tool outputs and generates highly explainable results.

2

Stress-Free & Parallel Processing

Immune to operational pressure, delivering efficient and parallel analysis.

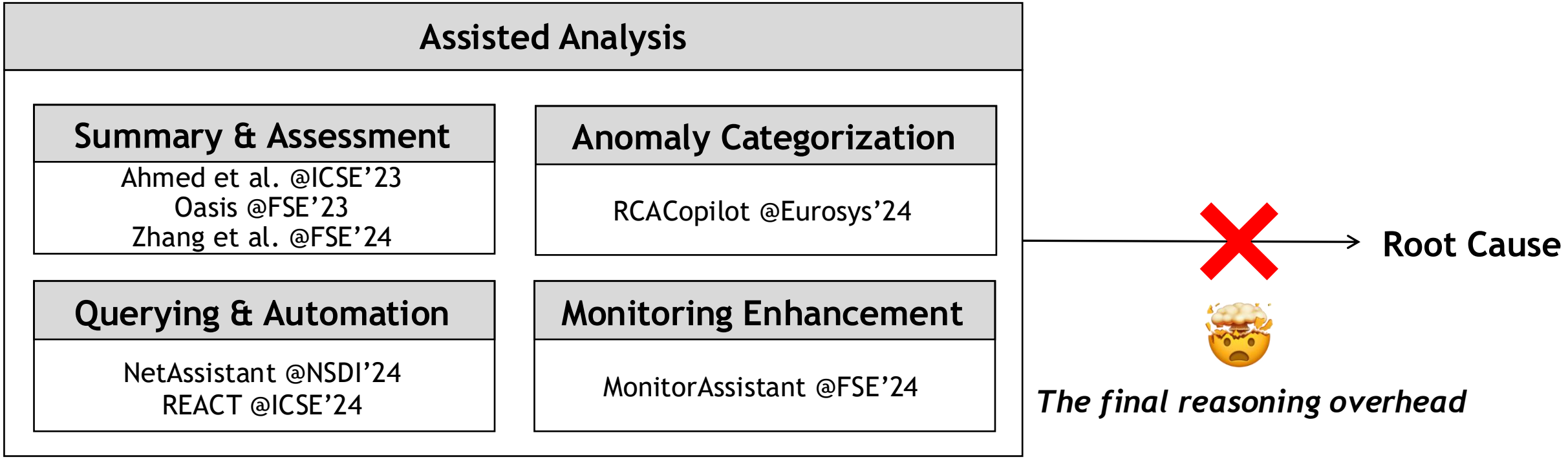
3

Versatility & Customizability

LLMs cover a wide range of problems out-of-the-box and can adapts to specific tasks via prompting and fine-tuning.

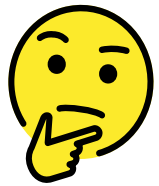
The last-mile problem of existing LLM-based works

Existing works are limited in two ways: (1) providing coarse-grained analysis; (2) relying on partial information.



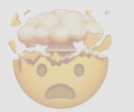
The last-mile problem of existing LLM-based works

Existing works are limited in two ways: (1) providing coarse-grained analysis; (2) relying on partial information.



Can we push the assistance one step further?

Querying & Automation	Monitoring Enhancement
NetAssistant @NSDI'24 REACT @ICSE'24	MonitorAssistant @FSE'24



The final reasoning overhead

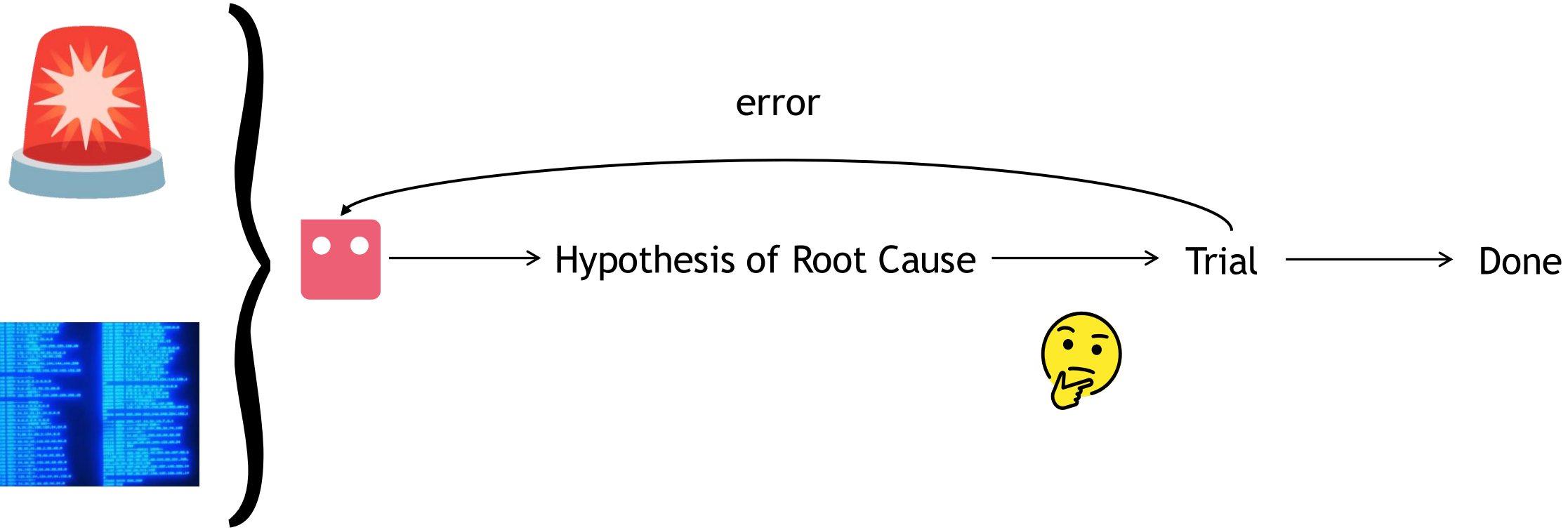
Our insights

1. We assist, not replace. Human operators are the final defense.
2. Validation is faster than computation.

   LLM proposes potential *hypotheses*, leaving humans with *trial and error*.

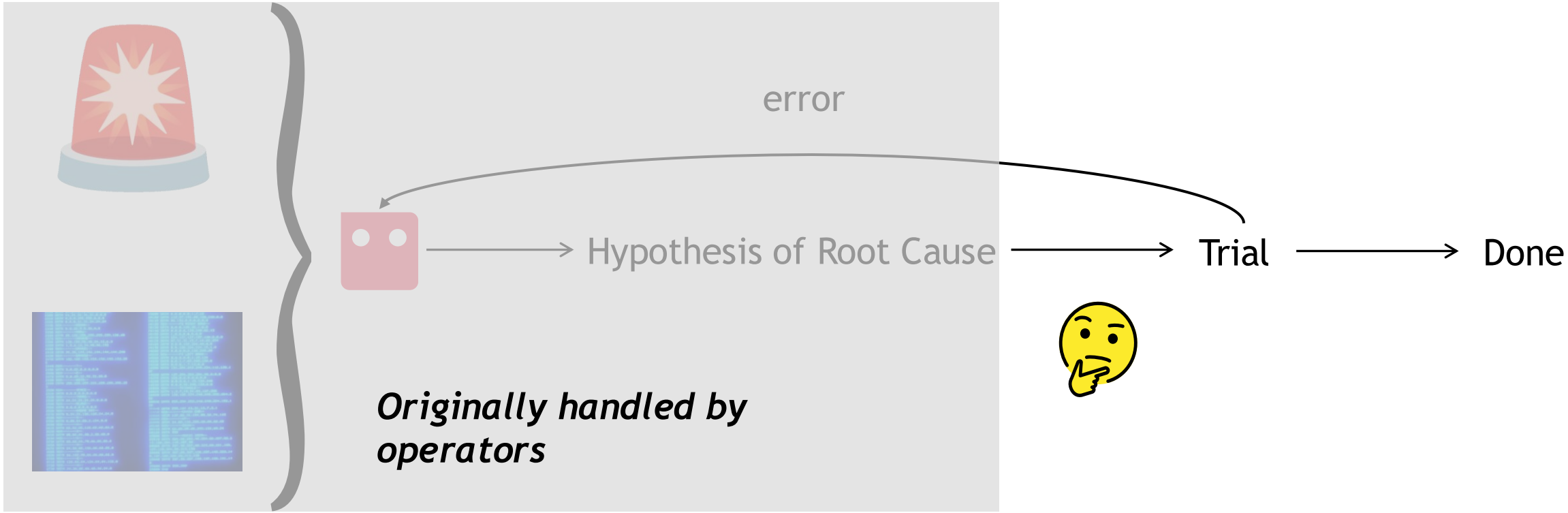
The ideal workflow

LLM reads and reasons about the logs, generating hypotheses to be validated.



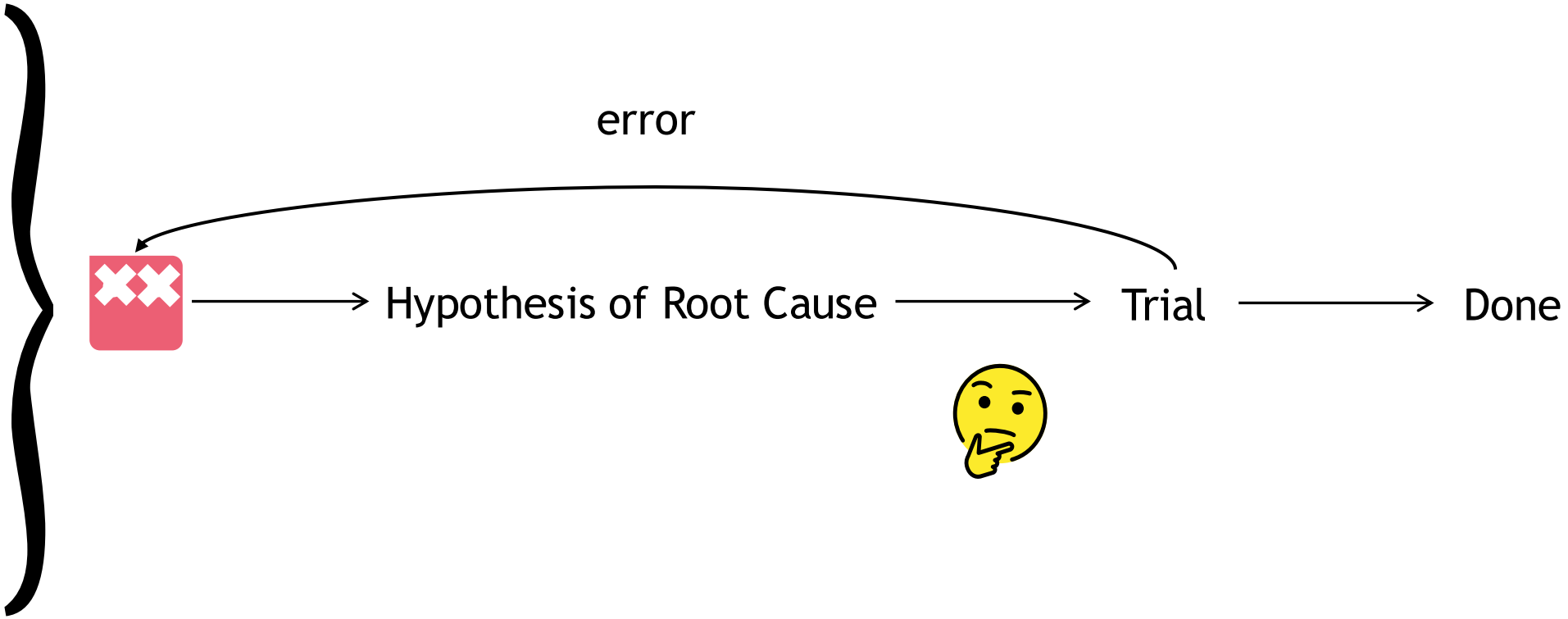
The ideal workflow

LLM reads and reasons about the logs, generating hypotheses to be validated.



Observation

Even for an LLM, the sheer volume of monitoring data can be overwhelming



Observation

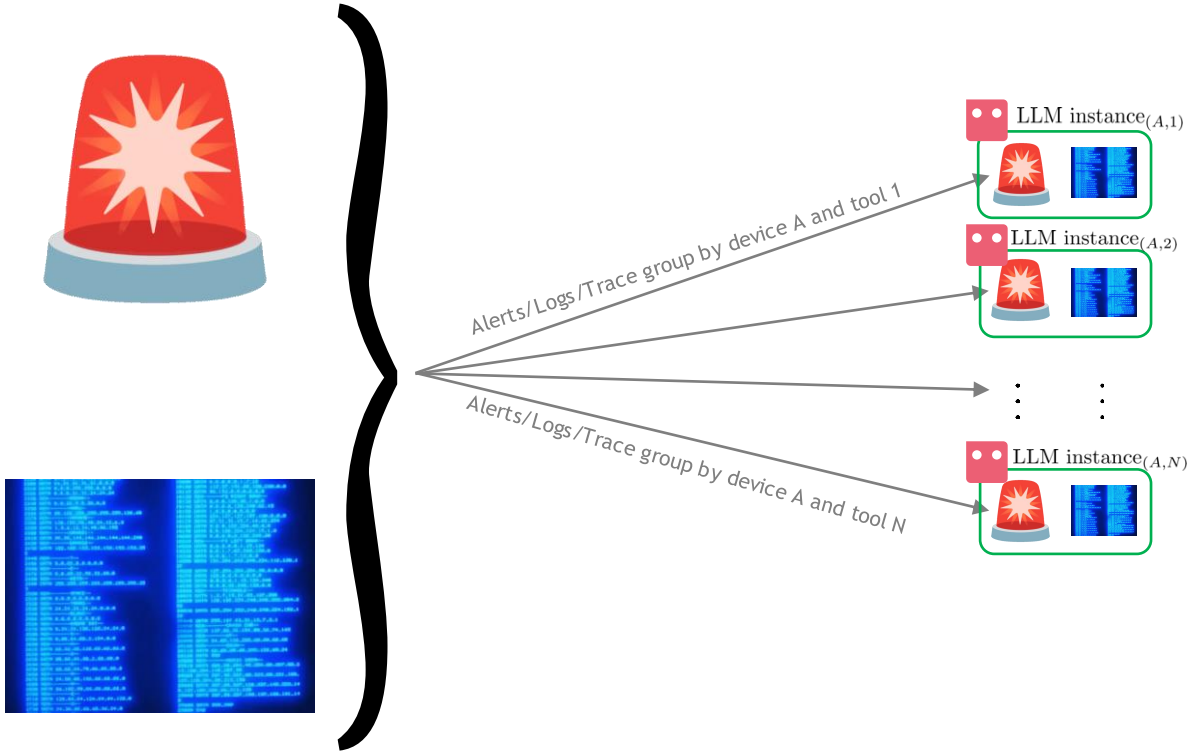
Even for an LLM, the sheer volume of monitoring data can be overwhelming

BIAN

a hierarchical, multi-pipeline framework for LLM-based failure localization designed to accelerate operator-led incident investigation.

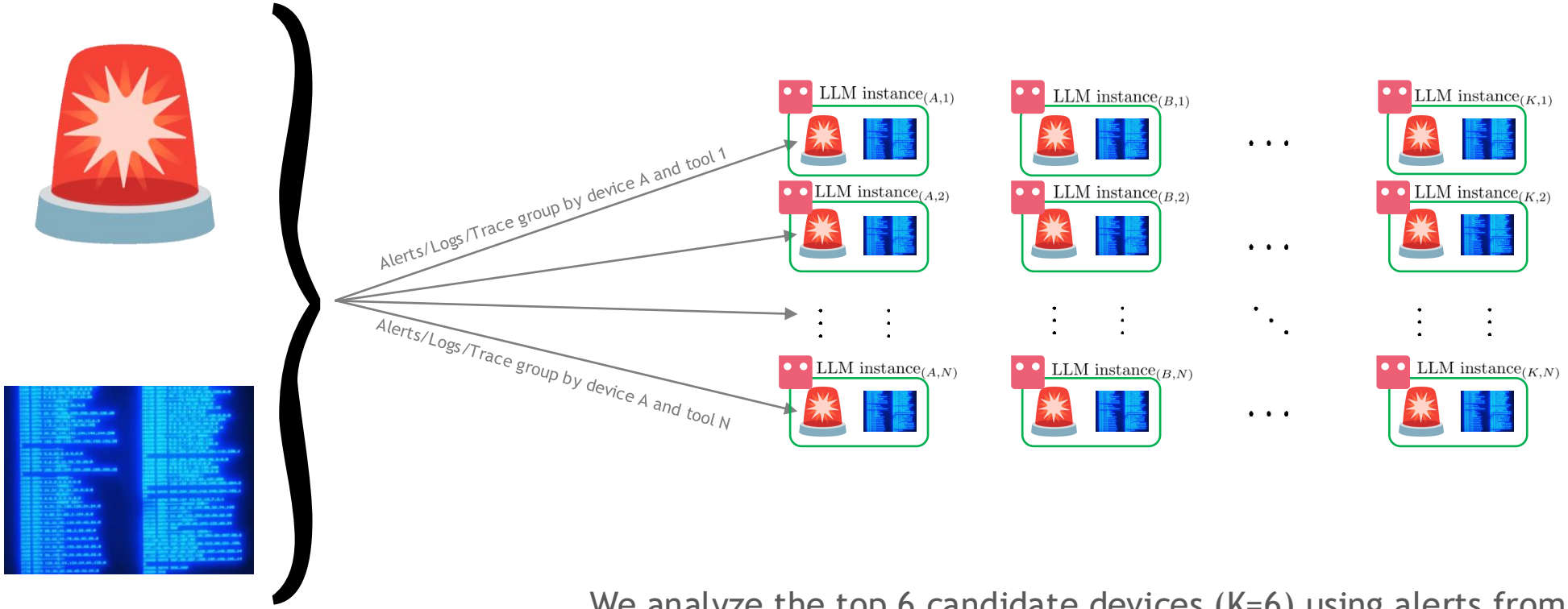
BIAN's Workflow

To overcome context overload, we scope each LLM to a narrow task: logs from one tool, for one device.



BIAN's Workflow

To overcome context overload, we scope each LLM to a narrow task: logs from one tool, for one device.



We analyze the top 6 candidate devices (K=6) using alerts from 11 different monitoring tools (N=11). See our paper for a detailed discussion.

BIAN's Design: Monitor Alert Summary

Each type of alert is processed by a LLM agent which is prompted to extract key information and summarize anomalous behaviors.

Prompt for Alert 1

Role definition
Input Field Description
Summary Guidelines
Alert Example
Expected Summary Format

Prompt for Alert 2

Role definition
Input Field Description
Summary Guidelines
Alert Example
Expected Summary Format

. . .

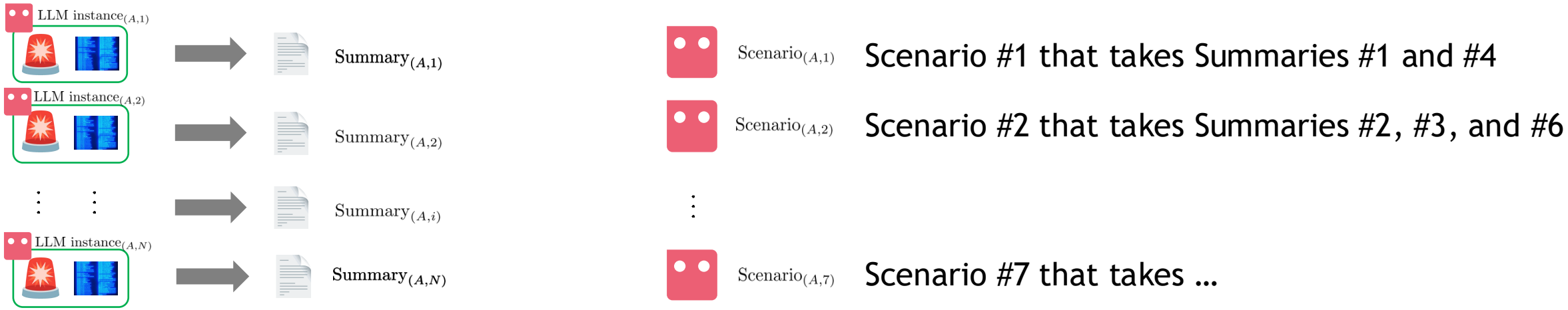
Prompt for Alert 11

Role definition
Input Field Description
Summary Guidelines
Alert Example
Expected Summary Format

We analyze the top 6 candidate devices (K=6) using alerts from 11 different monitoring tools (N=11). See our paper for a detailed discussion.

BIAN's Workflow

For each device, relevant alert summaries are selected and fed into specialized LLM agents to perform single-device anomaly analysis.



We use 7 dedicated LLM agents, one for each of the predefined anomaly scenarios. See our paper for a detailed discussion.

BIAN's Design: Device Anomaly Analysis

The analysis is driven by Standard Operating Procedures (*SOPs*), which can deduce fundamental anomaly features from the monitoring reports.

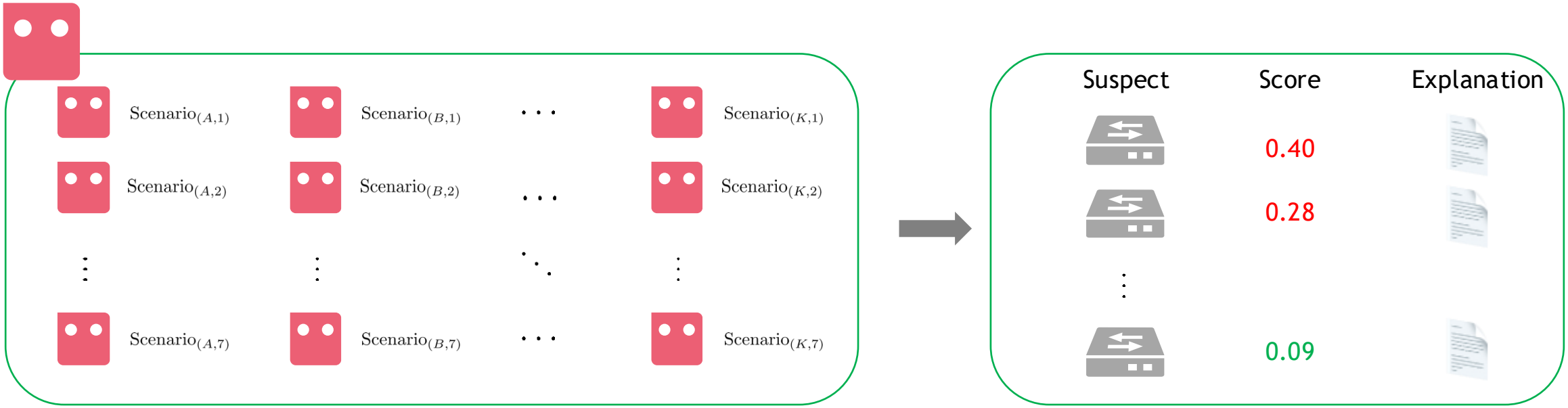
SOP - Traffic Drop Scenario

1. **Access the alarm dashboard of the monitoring system and check if there are any "Ping Unreachable" or "Host Down" alerts.**
2. **Check …**
3. **If any of the checkpoints above are confirmed to be true, the "Traffic Drop" anomaly scenario is considered confirmed.**

This analysis focuses on observable failures. Issues that do not generate alerts, such as silent packet drops, are out of BIAN's current scope. See our paper for a detailed discussion.

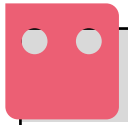
BIAN's Workflow

BIAN consolidates the anomaly analysis reports from all suspect devices to reason about the root cause.



BIAN's Design: Joint scoring

The LLM agent is prompted to reason about the evidence, weighing both the frequency and severity of each device's anomalies.



Prompt for Joint Scoring

 # Role definition

 # Scenario Anomaly Analysis Reports

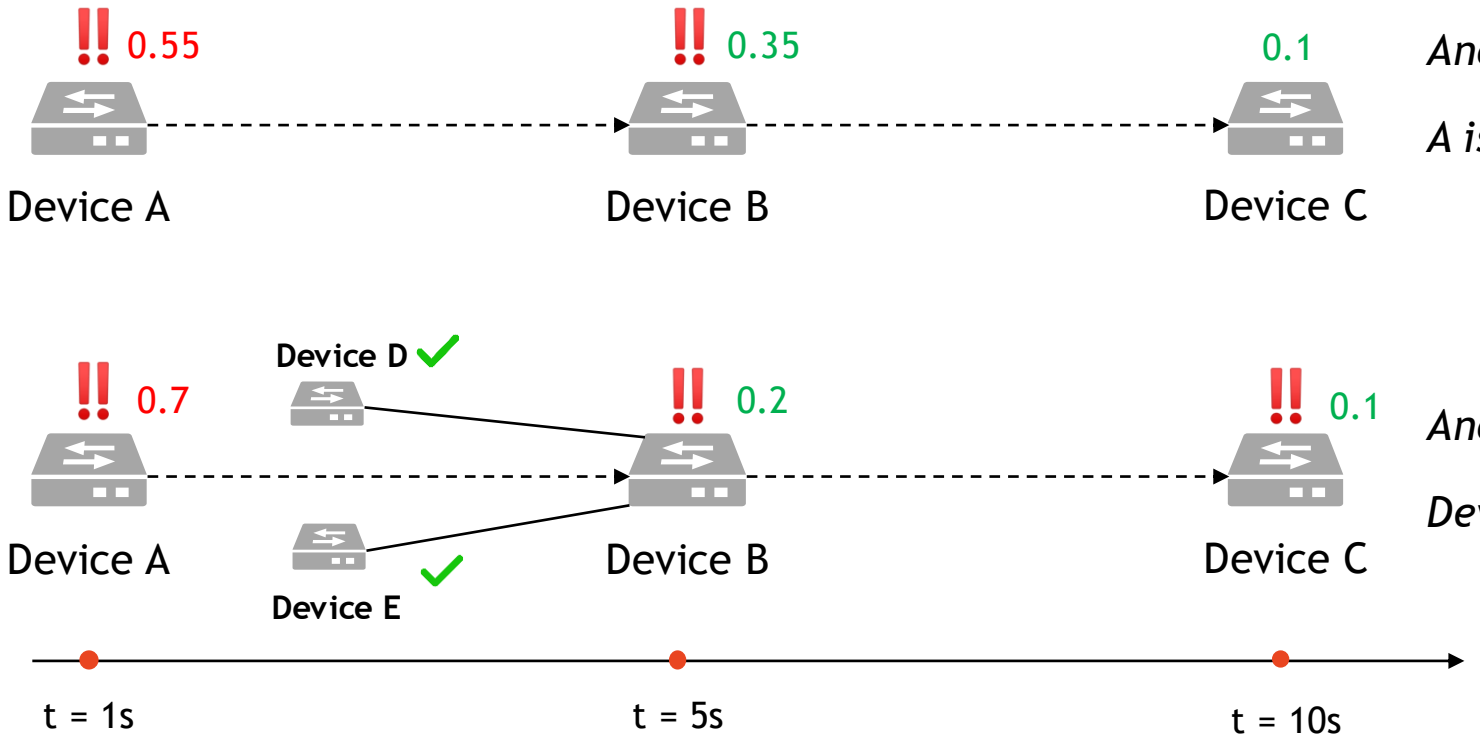
 # SOP of Investigation

 # Example

 # Expected Analysis Format

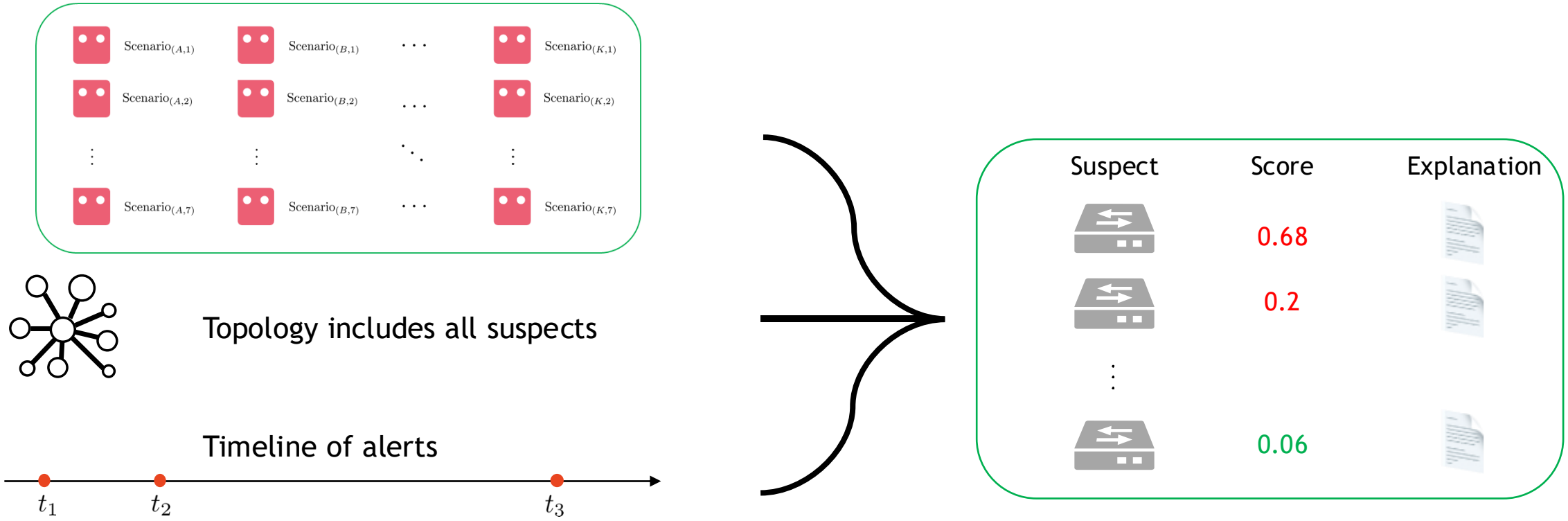
Observation: Anomalies are rarely isolated

The spatiotemporal context is a powerful semantic signal for causality. Failures tend to propagate across devices (spatial) and evolve over time (temporal).



BIAN's Design: Three-pipeline Integration

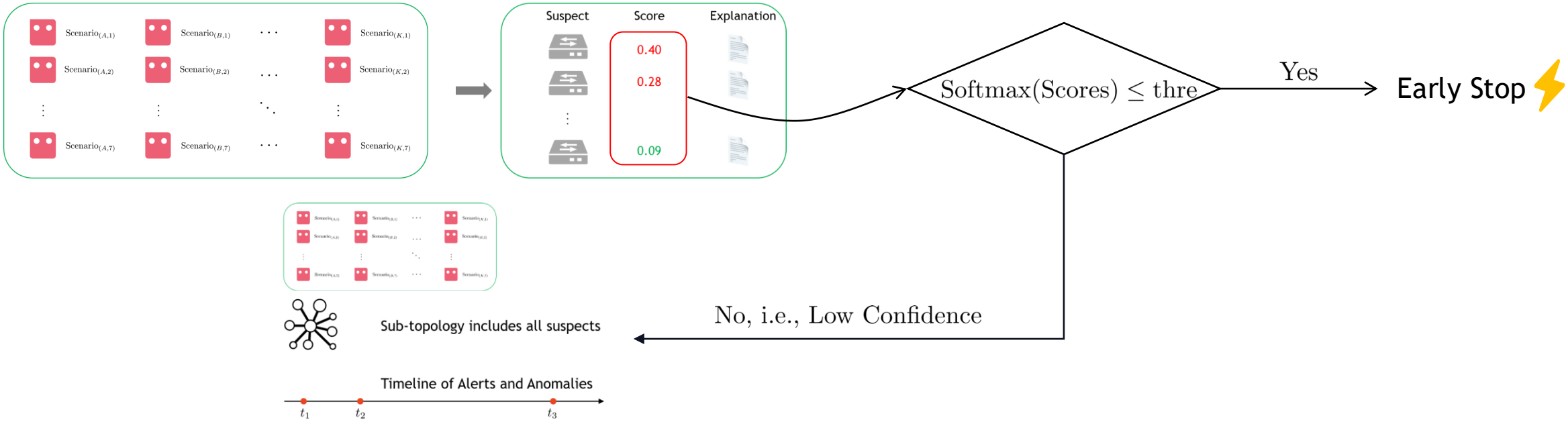
To further enhance the robustness of this analysis, we enrich it with additional semantic context through three-pipeline integration.



The network topology, covering all suspect devices and their structural relationships, is generated with a dedicated algorithm. See our paper for a detailed discussion.

BIAN's Design: Early Stop

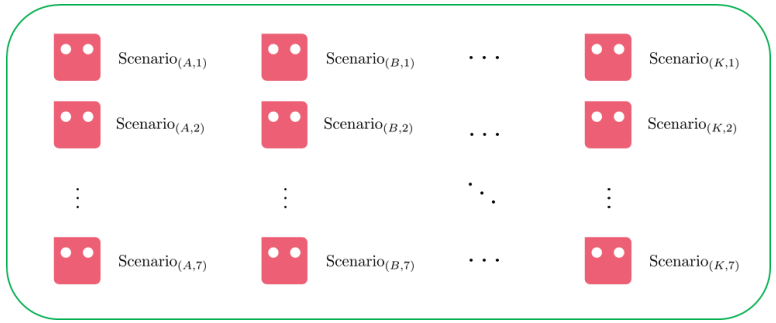
The integrated solution can be overkill for simpler cases. We introduce an early-stop mechanism, triggered by a high-confidence initial assessment.



Activate the full three-pipeline integration.

BIAN's Design: The "Rank of Ranks" Mechanism

To tackle LLMs' inherent randomness, "Rank of Ranks" averages the outputs from multiple reasoning trials.



Analysis Task

Trial 1	Suspect	Score	Rank
	A	0.6	1
	B	0.3	2

Trial 2	Suspect	Score	Rank
	A	0.5	1
	B	0.4	2

Trial 3	Suspect	Score	Rank
	A	0.2	2
	B	0.7	1

LLM Randomness Outlier

Device	Avg. Score $= \sum_{i=1}^{3} S_i$
B	0.46
A	0.43

Device	Avg. Rank $= \sum_{i=1}^{3} R_i$
A	1.33
B	1.67

System Optimizations

To ensure our design is practical for real-world deployment, we introduce some key optimizations for both model specialization and execution efficiency.

1

Fine-tuning

Goal: Train smaller, more specialized models for higher speed.

Dataset: Synthetic data generated based on real-word distribution and partially validated.

Application: alert summary, anomaly analysis.

2

Parallel Execution

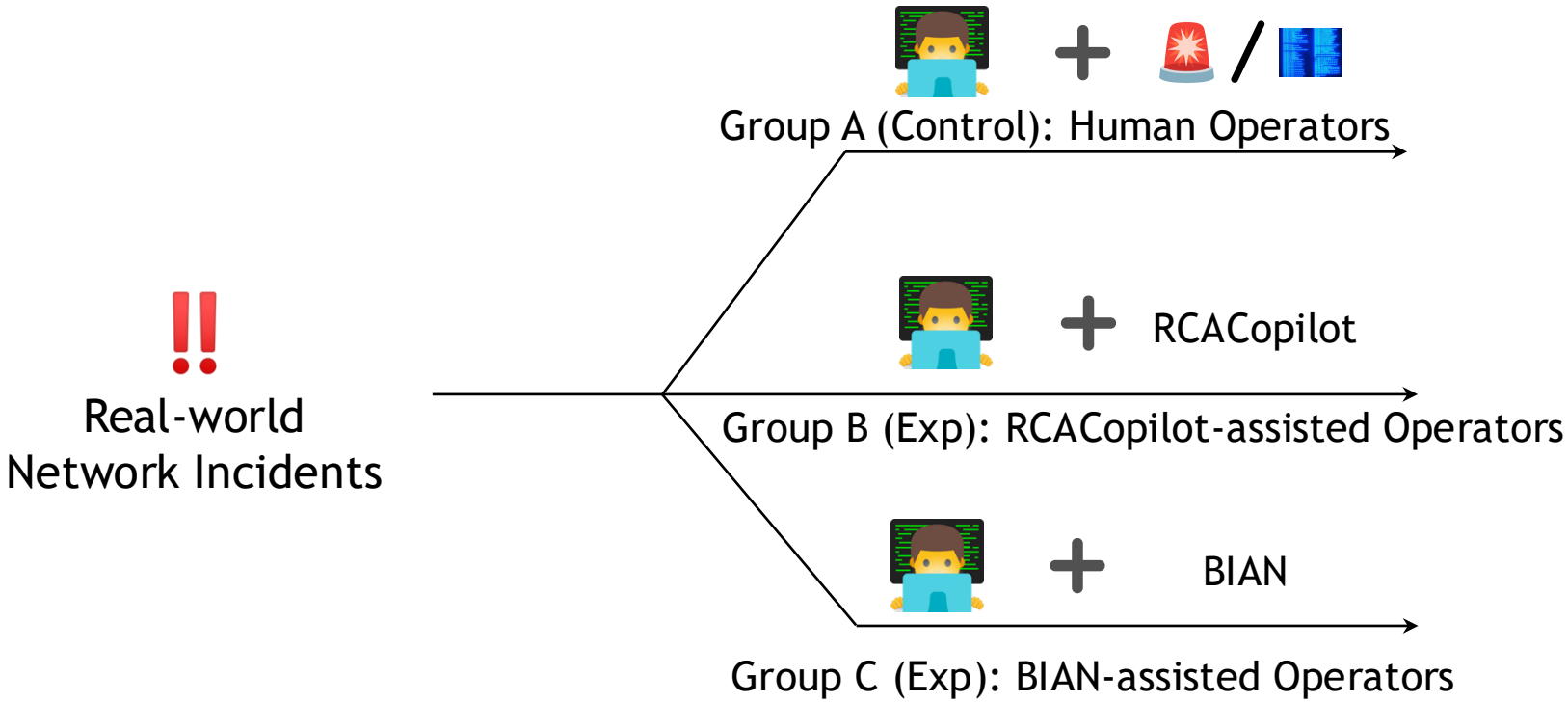
Principle: Execute all non-interdependent agents concurrently.

Key Use Cases:

- **Agents in Pipeline 1 (alert summary, etc.)**
- **Multiple runs for Rank of Ranks.**

Evaluation Setup: A/B Testing with Human Operators

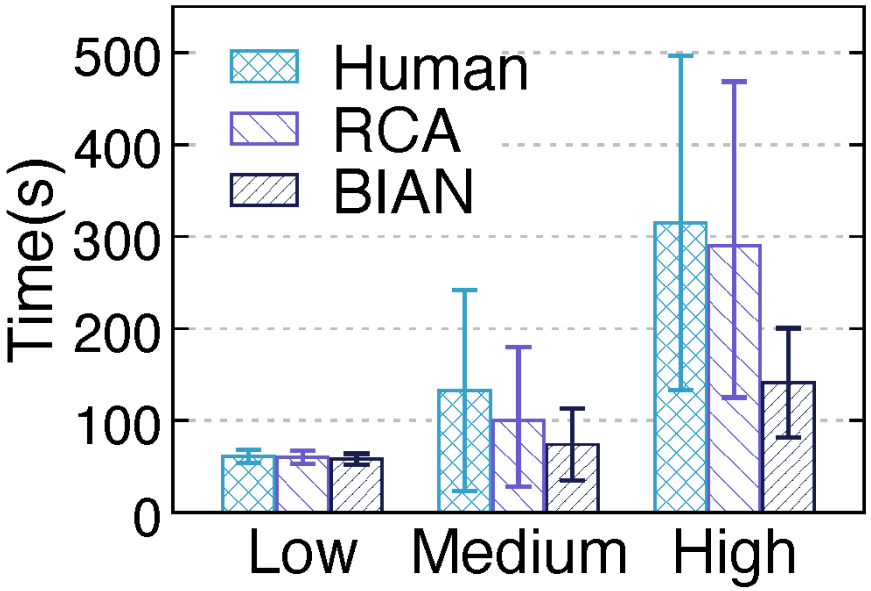
To measure BIAN's real-world impact on operator efficiency, we designed a A/B testing framework using a 'shadow operation' model.



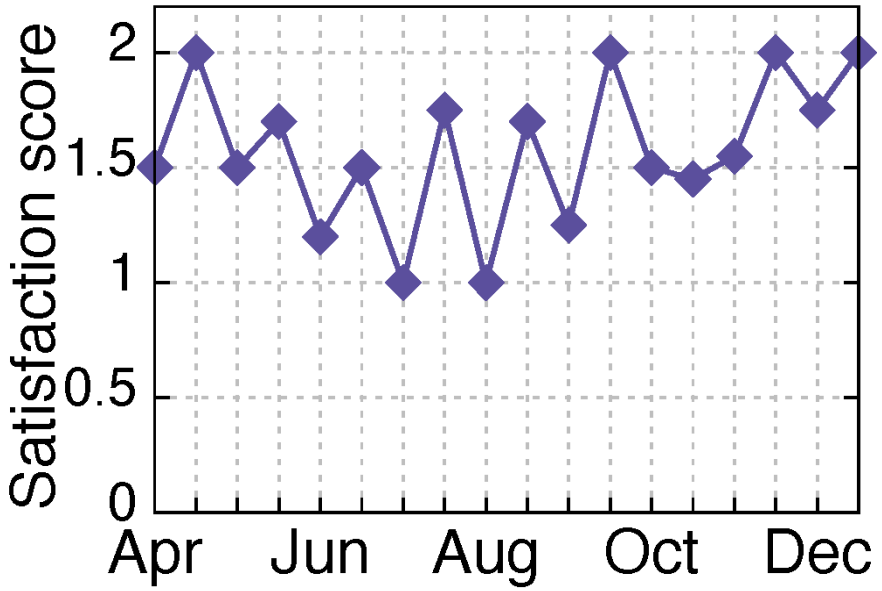
Key Performance Indicators
Time to Root Cause Identification (TTR)
Accuracy of Diagnosis
Operator Confidence / Feedback

Evaluation

BIAN significantly reduces troubleshooting time, especially in complex cases, while its generated explanations consistently earn high satisfaction from human operators.



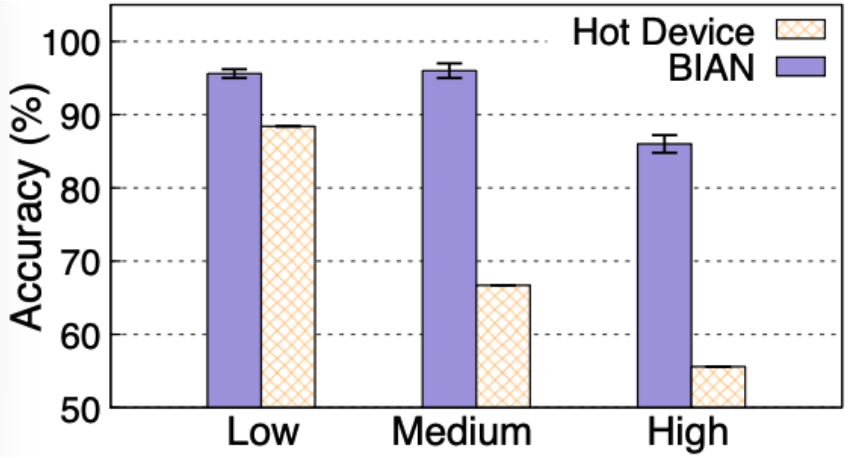
TTR of solo and assisted incident investigations.

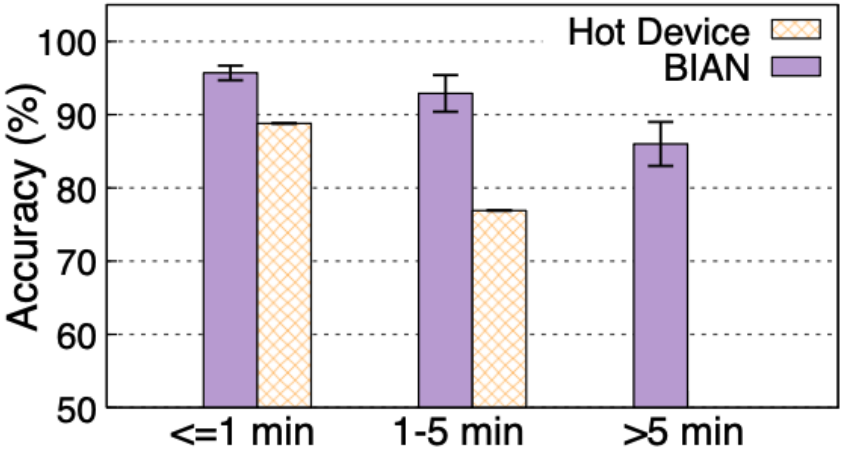


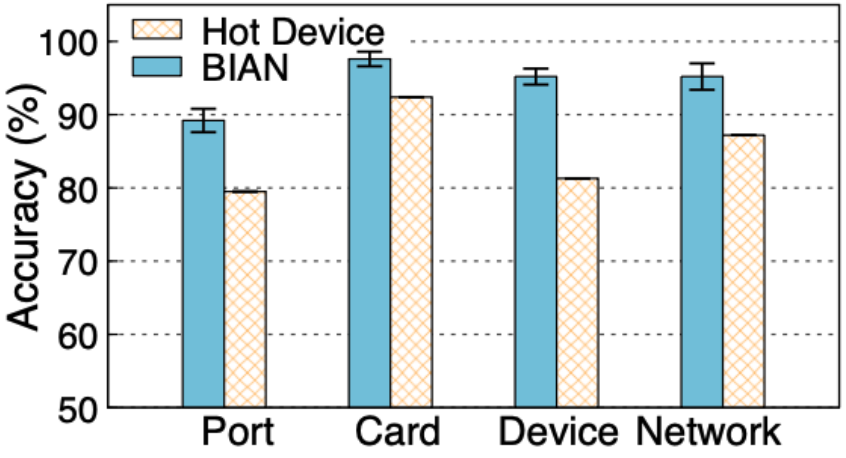
Operators' satisfaction with BIAN's explanations.

Evaluation

We compare the accuracy of BIAN and Hot Device. Accuracy is defined as whether the system picks the actual error device as top-1.

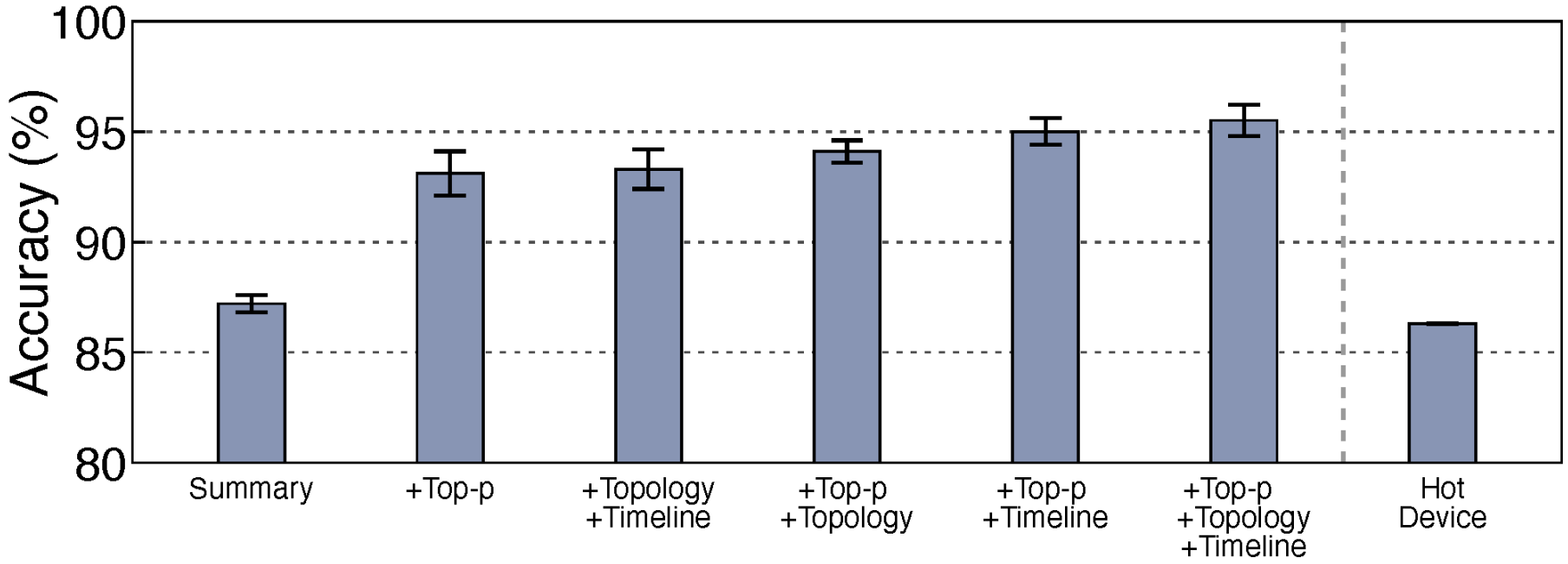






Evaluation

To validate our design choices, we perform an ablation study to quantify the contribution of each key component to the final accuracy.



Note: Top-p is a technique to improve reasoning performance. For a detailed discussion, please see our paper.

Conclusion

We presented BIAN, a framework demonstrating that LLMs can successfully facilitate failure localization in a live, large-scale production network.

- Hierarchical reasoning
- Multi-pipeline integration
- Other enhancements: continuous updating, early stop, …

Thank you!