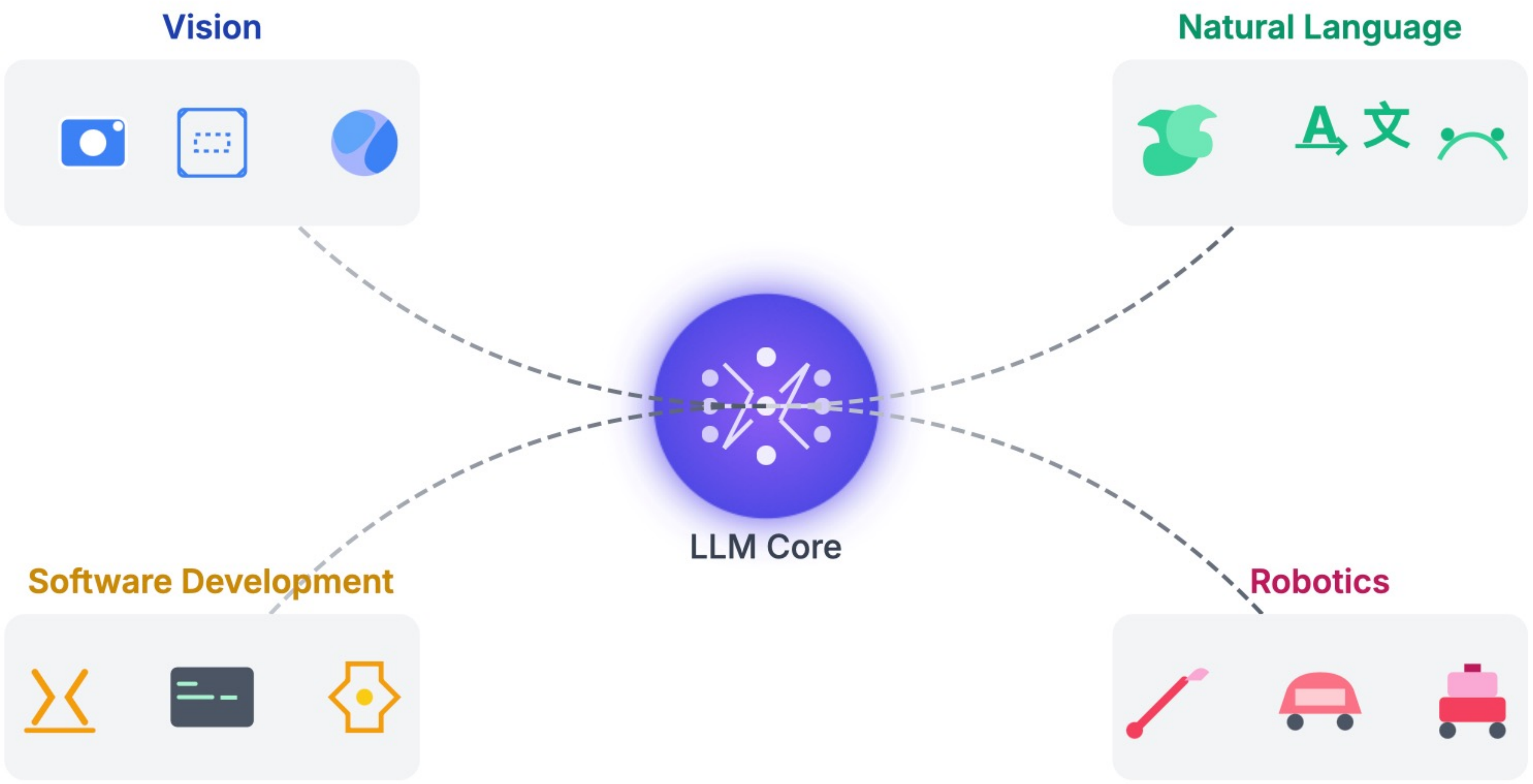
L4: Diagnosing Large-scale LLM Training Failures via Automated Log Analysis

Zhihan Jiang[1], Junjie Huang[1], Guangba Yu[1], Zhuangbin Chen[2], Yichen Li[1], Renyi Zhong[1]
Cong Feng[3], Yongqiang Yang[3], Zengyin Yang[3] and Michael R. Lyu[1]

[1]*The Chinese University of Hong Kong,*
[2]*Sun Yat-sen University,*
[3]*Huawei Cloud*

Background

Large Language Models (LLMs) are everywhere for everyone



Scaling Law in LLMs

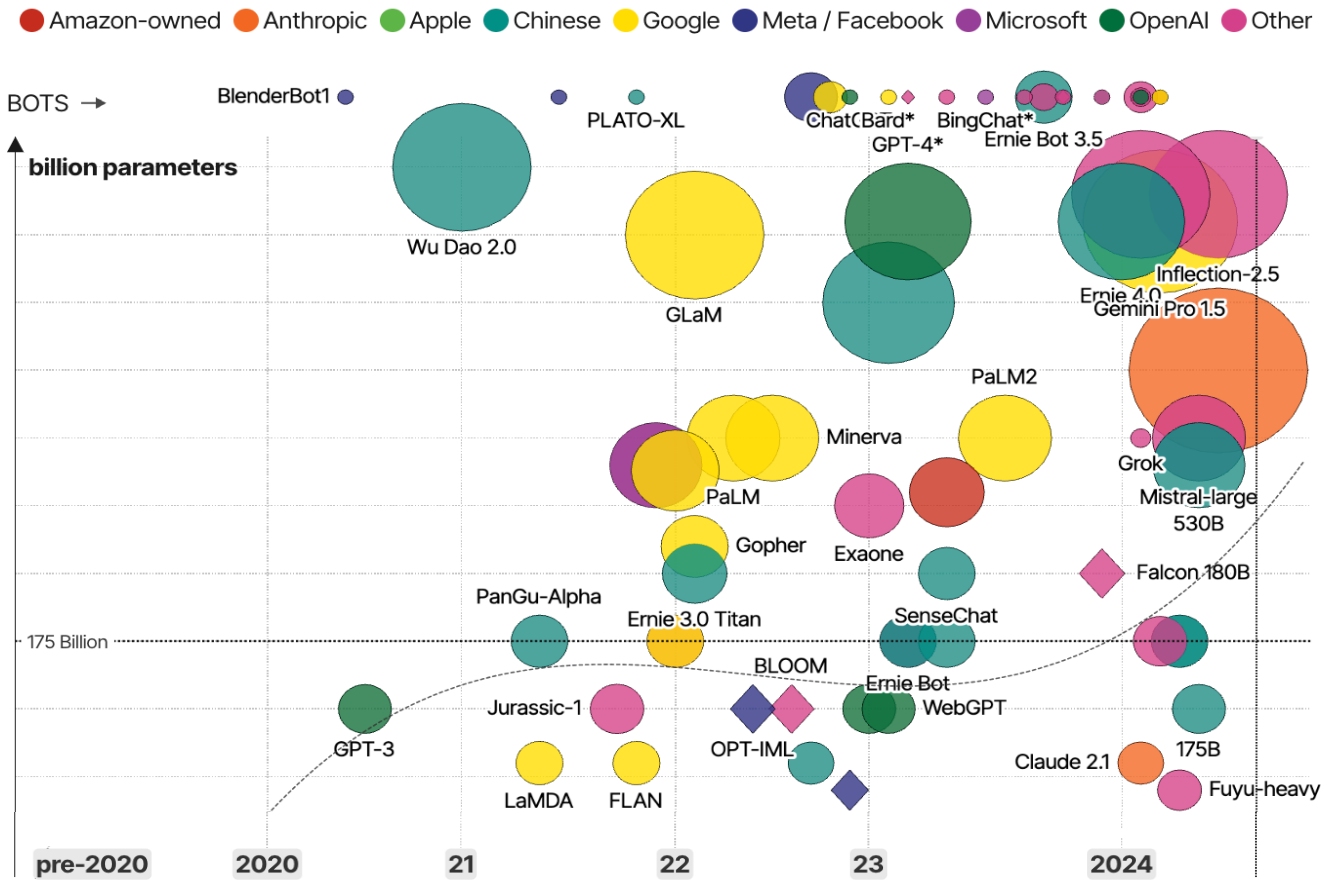
LLMs have grown rapidly in scale, e.g., >100B parameters

MISTRAL AI_ Mistral Large 123B

OpenAI GPT-3 175B

Meta LLaMA3.1 405B

deepseek DeepSeek-R1 671B



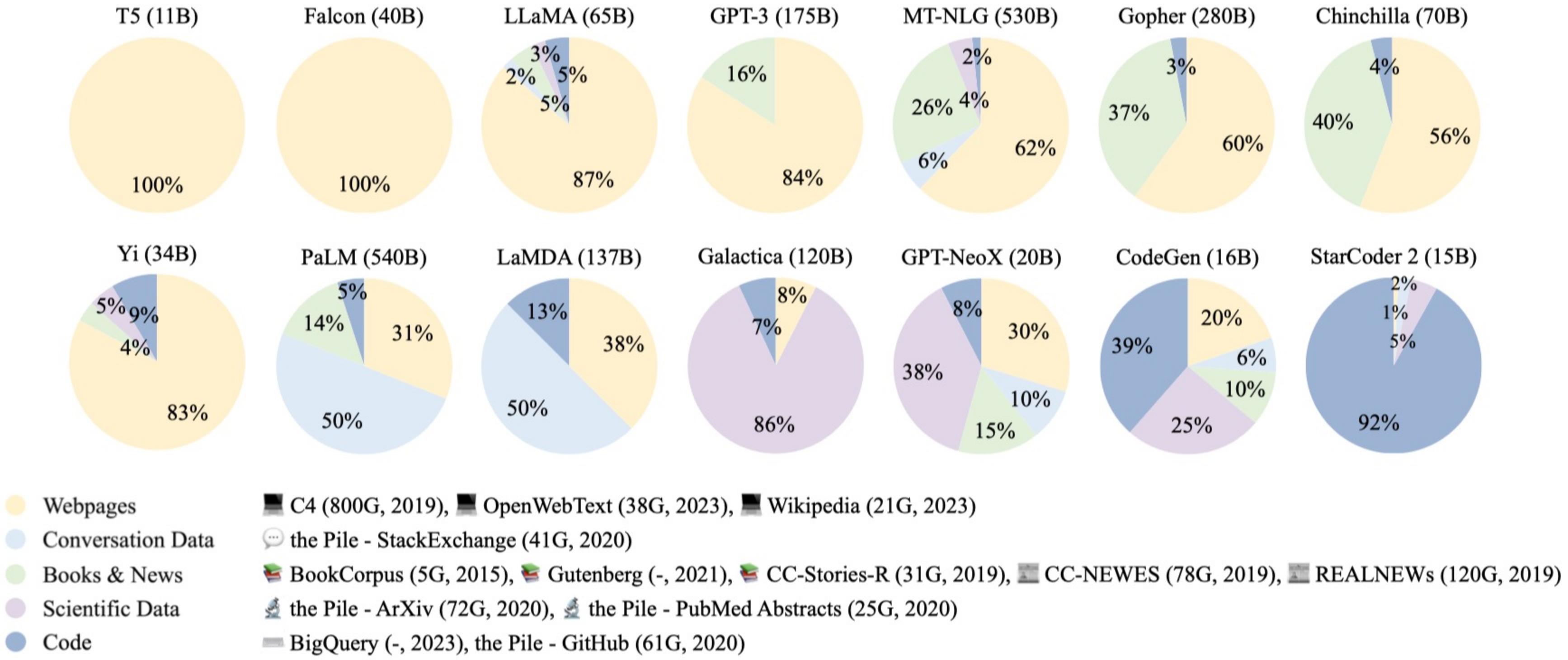
https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/

Scaling Law in LLMs

The training datasets are also increasingly diverse and massive

- **LAION-2B-en** datasets: over 360B tokens

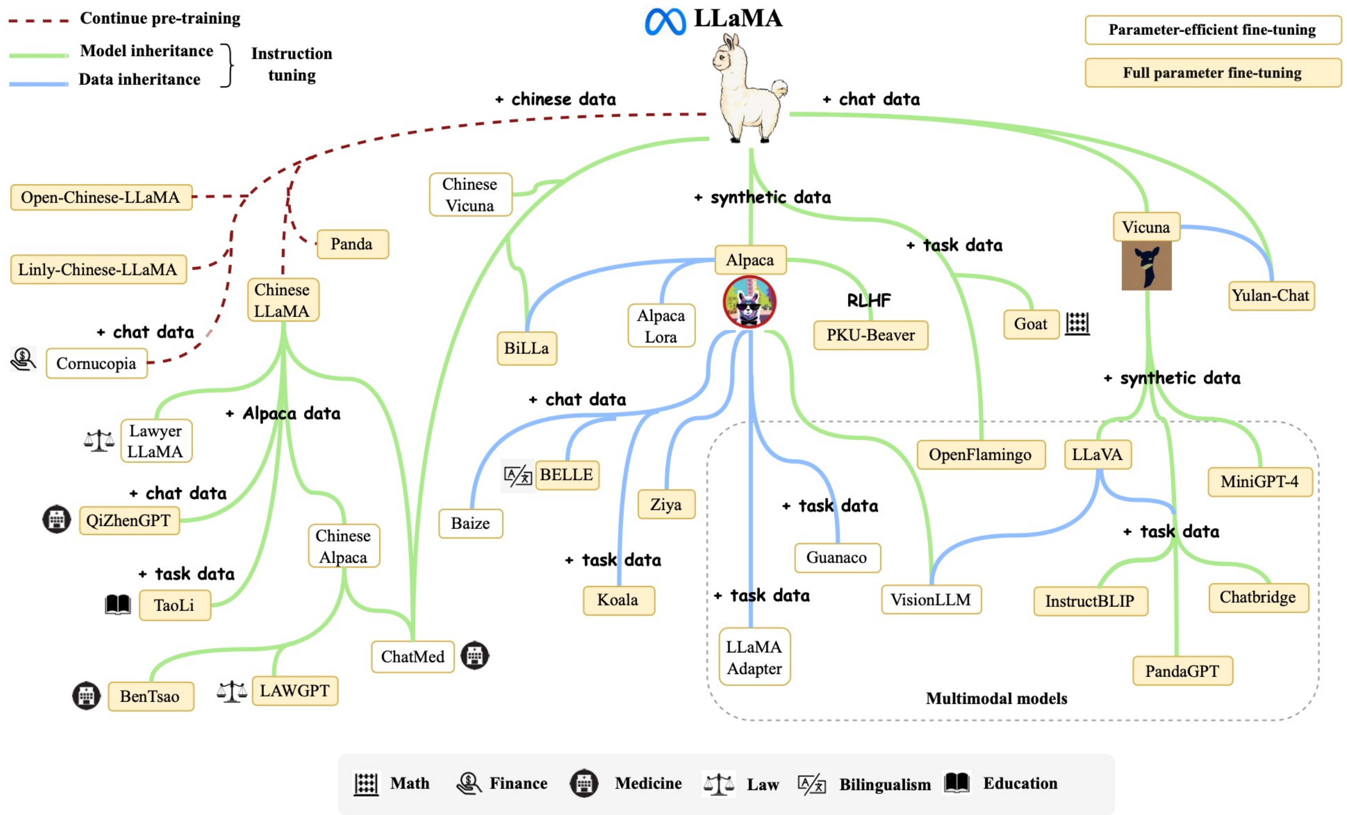
- **RedPajama** datasets: over 30T tokens



A survey of large language models[J]. arXiv preprint arXiv:2303.18223, 2023.

LLM Training and Tuning

Fine-tuning LLMs on domain-specific datasets is essential for achieving SOTA performance in specialized fields



A survey of large language models[J]. arXiv preprint arXiv:2303.18223, 2023.

LLM Training and Tuning

Training or tuning LLMs require substantial computing resources

Full-training

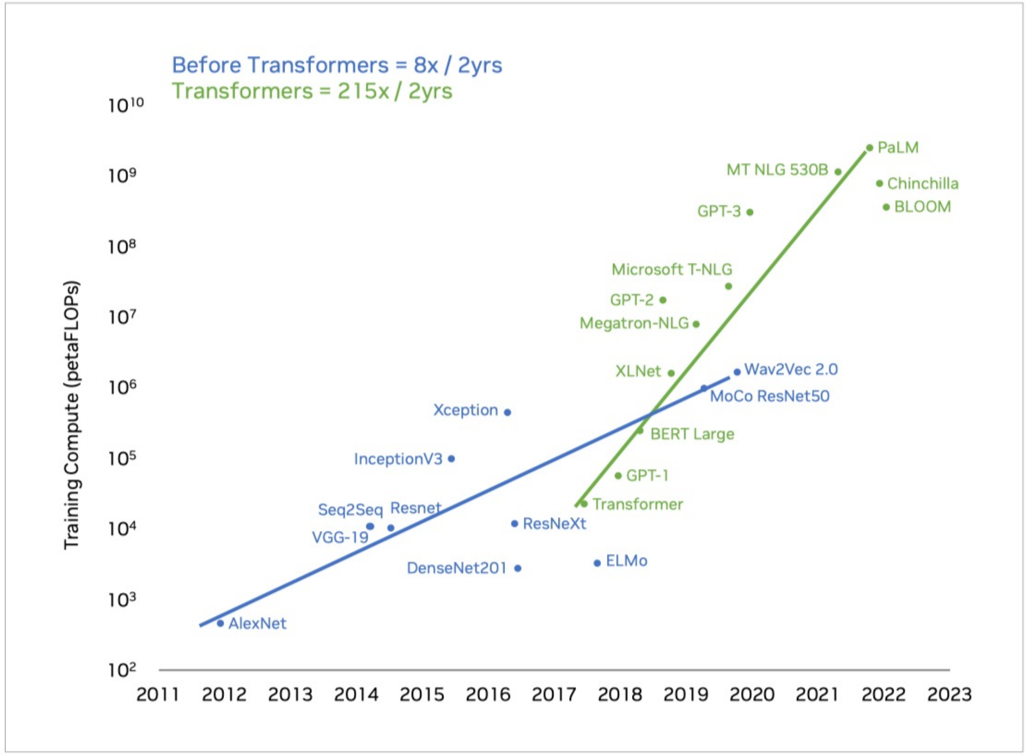
BLOOM-176B: 384 A100 GPUs for 3.5 months

LLaMA 3.1 405B: 16,384 H100 GPUs for 54 days

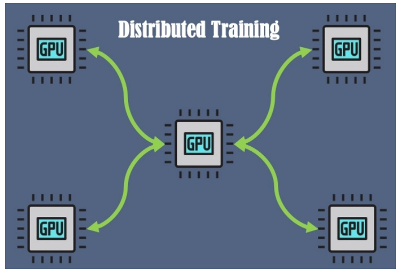
DeepSeek-V3-671B: 2048 H800 GPUs for 56+ days (2.788M GPU Hours)

Post-training (Fine-tuning)

DeepSeek-V3-671B: 64 H800 GPUs for 3+ days (5K GPU Hours)



LLM Training Reliability

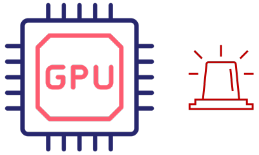


As the scales of LLMs and the training resources grow, the likelihood of failures during the training process increases significantly

The core reason: synchronization properties during the distributed training process

A local fault in a specific component (e.g., a GPU), can disrupt the entire training process

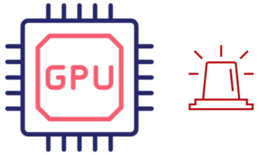
probability=0.1%



1 GPU per day

scaling up

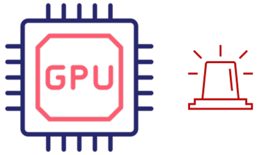
probability=1 - (1 - 0.1%) ^ 128 = 12%



128 GPUs per day

scaling up

probability=1 - (1 - 0.1%) ^ 8192 = 99.97%



8192 GPUs per day

LLM Training Reliability

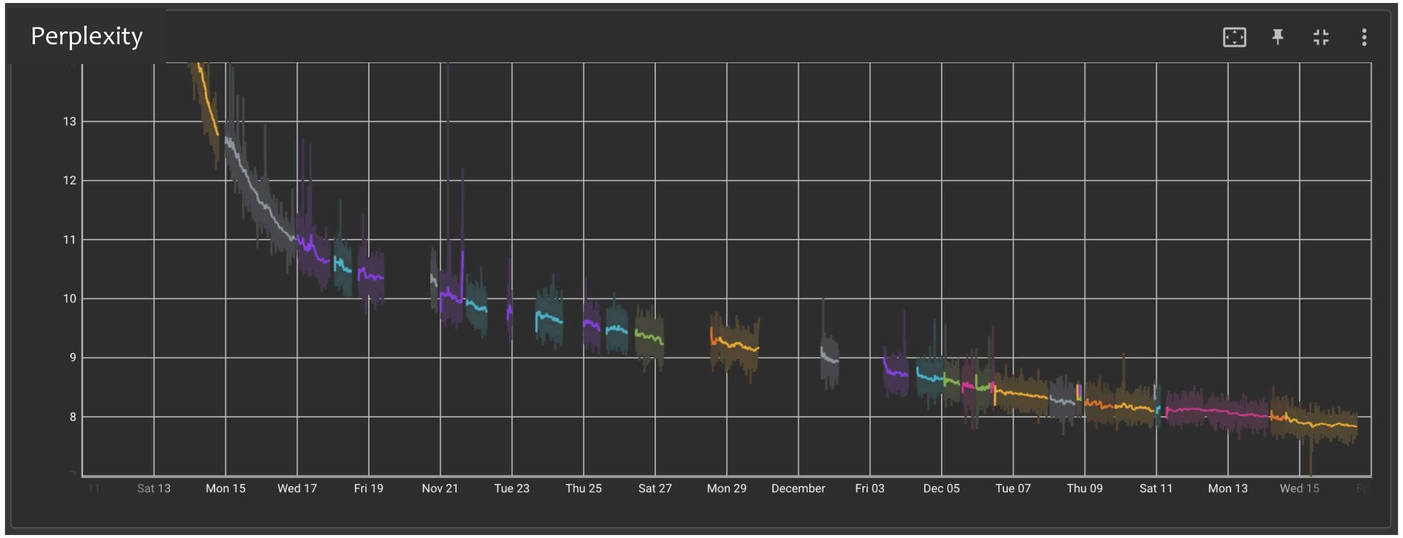
As the scales of LLMs and the training resources grow, the likelihood of failures during the training process increases significantly.

Example: Development life-cycle of OPT-175B from Oct. 2021 to Jan. 2022

35+ manual restarts

70+ automatic restarts

100+ cycling hosts

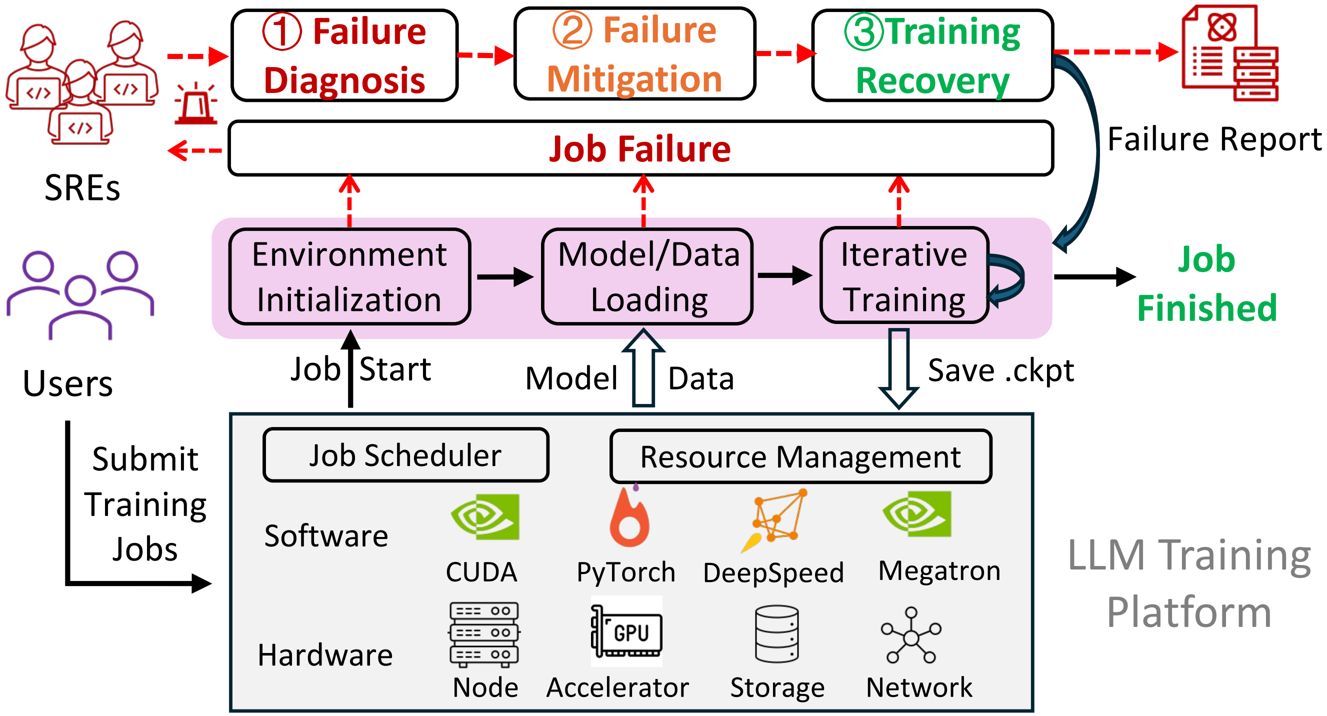


OPT: Open Pre-trained Transformer Language Models (from Meta)

LLM Training Platforms

LLM training platforms: simplify the LLM training and tuning process

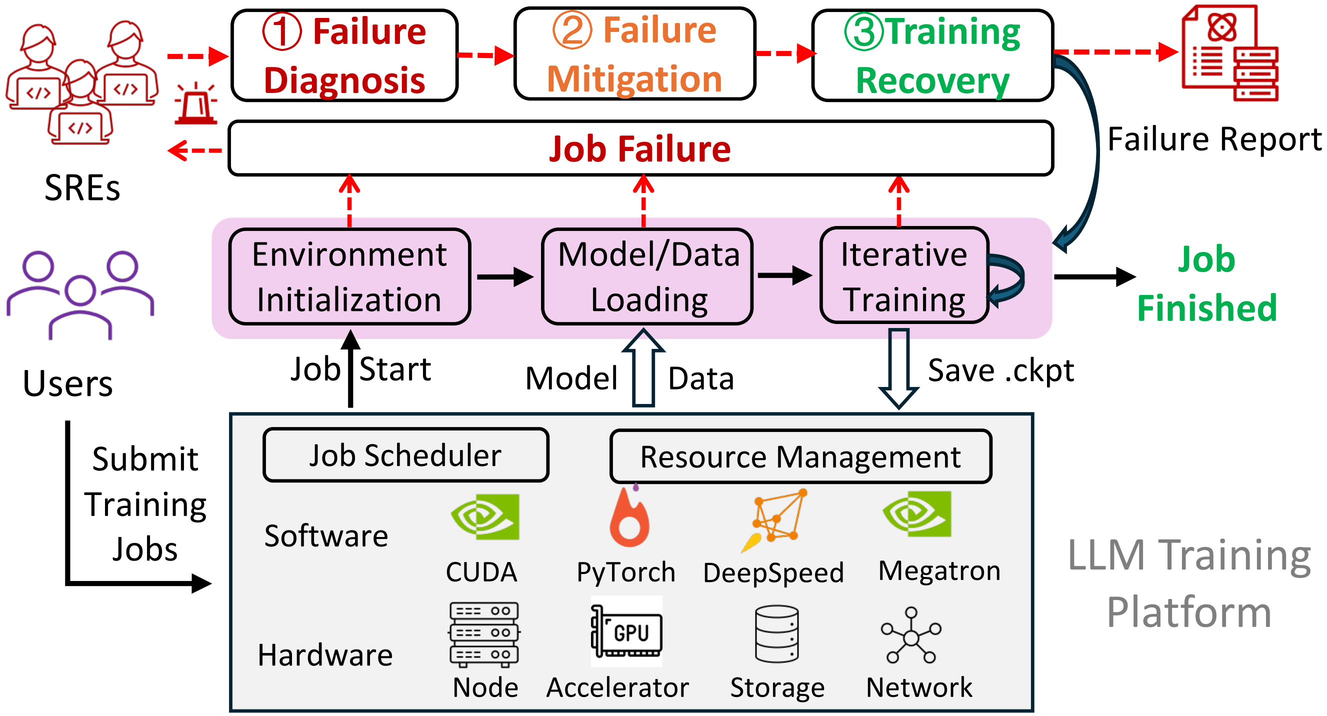
- Amazon SageMaker
- Google Vertex AI
- Databricks
- Huawei ModelArts



Our Production Platform-X

LLM Training Failure Diagnosis

Rapid and accurate failure diagnosis is critical for failure management



Our Production Platform-X

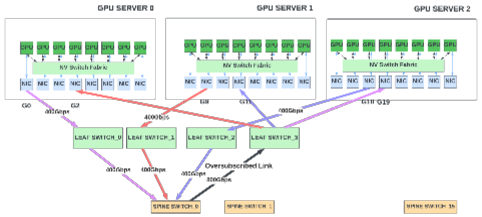
LLM Training Failure Diagnosis

Rapid and accurate failure diagnosis is critical for failure management

Challenges:

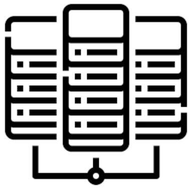
Horizontal:
- large-scale clusters and a multitude of components

High density computing servers



High performance networks

Distributed storages



AI Cluster

10K+	150+	60K+	1M+
GPU/NPU	Racks	Ports	Devices

Larger infrastructure != More computing power

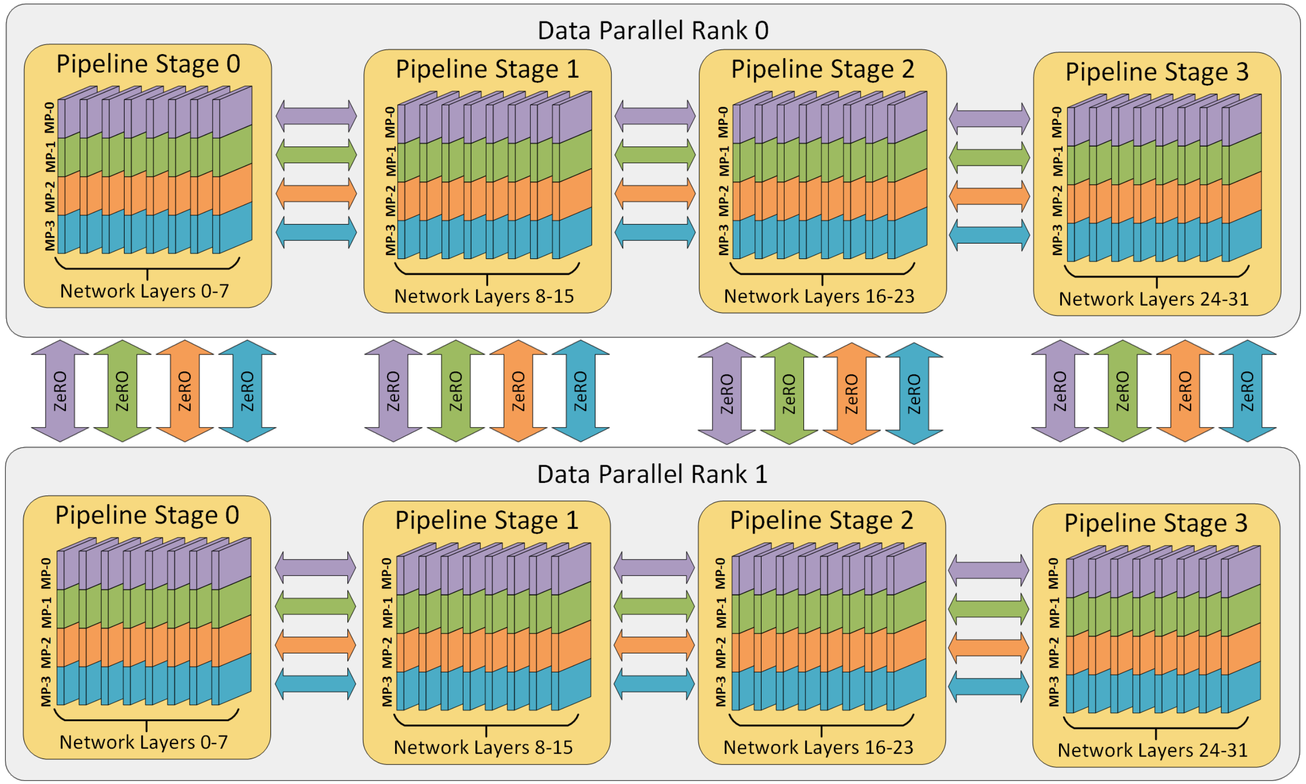
LLM Training Failure Diagnosis

Rapid and accurate failure diagnosis is a critical step in failure recovery

Challenges:

Horizontal:
- large-scale clusters and a multitude of components

Vertical:
- complex configurations and multi-layered architectures



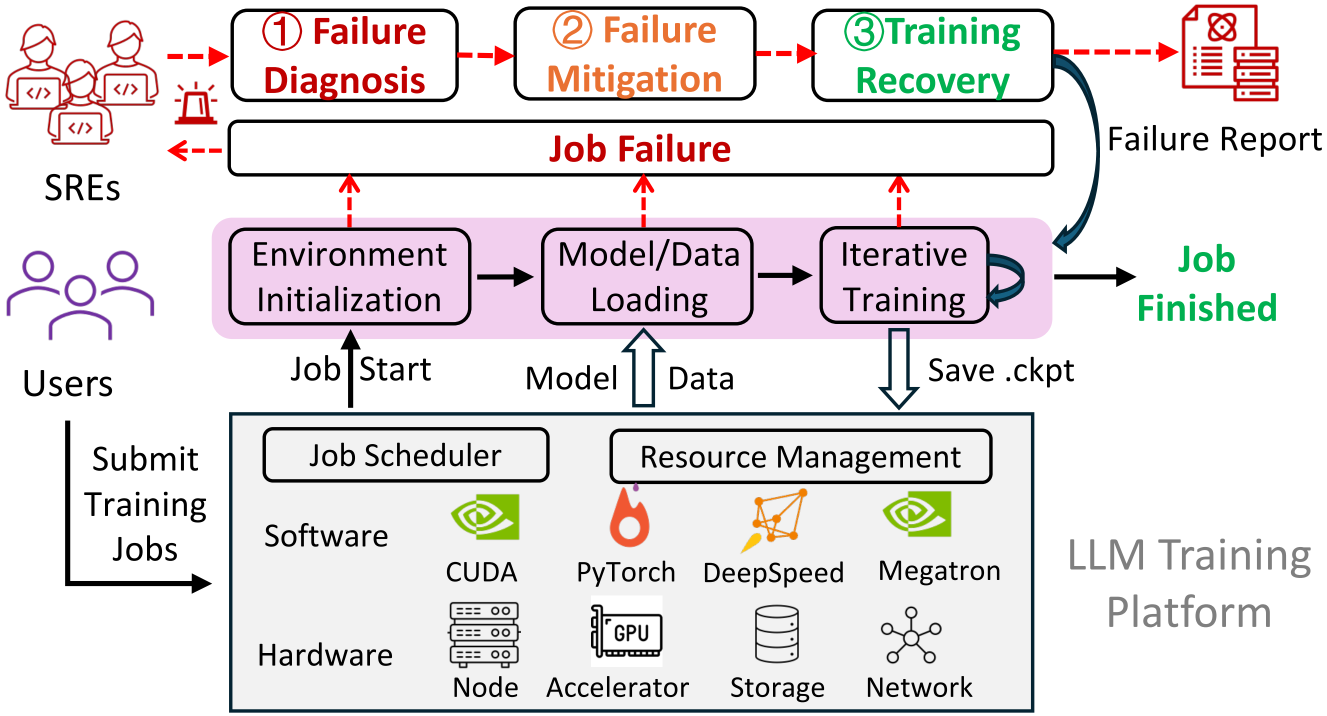
LLM Training Failure Diagnosis

Rapid and accurate failure diagnosis is a critical step in failure recovery

Challenges:

Horizontal:
- large-scale clusters and a multitude of components

Vertical:
- complex configurations and multi-layered architectures



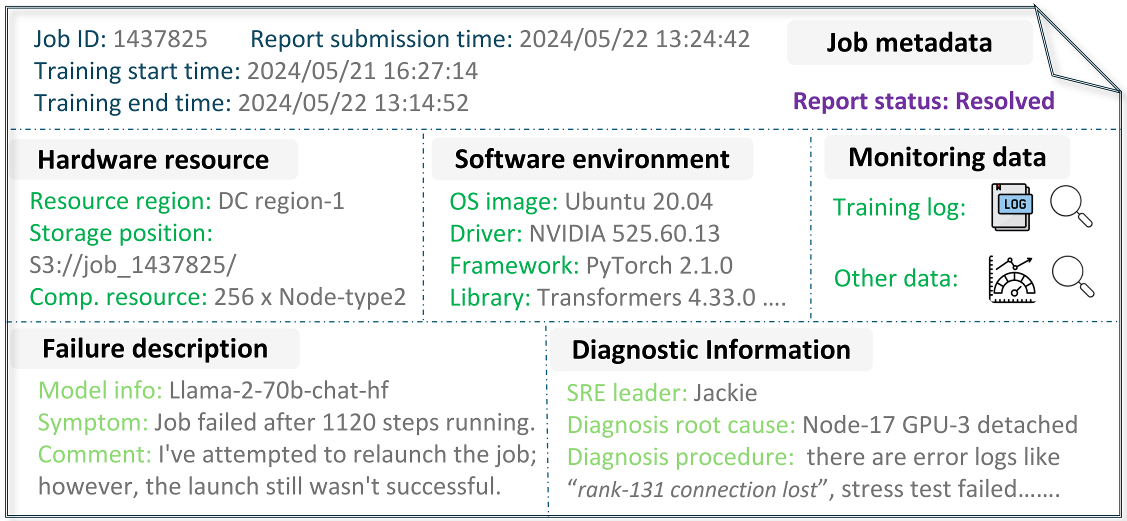
Our Production Platform-X

Understanding failures in LLM training and exploring opportunities for automated diagnosis are important!

Empirical Study on LLM Training Failures

Study Platform

Our multi-tenant production LLM training Platform-X supporting hundreds of internal users and partner companies



Study Subject

Collected 428 failure reports of failed large-scale LLM training jobs in Platform-X from May 2023 to April 2024

- average model size: **72.8B parameters**
- average accelerators: **941 GPUs or NPUs**

Study RQs

- RQ1: What are the common symptoms of LLM training failures?
- RQ2: What are the common root causes of LLM training failures?
- RQ3: What monitor data sources are typically used to diagnose LLM training failures?

RQ1: LLM training failures symptoms

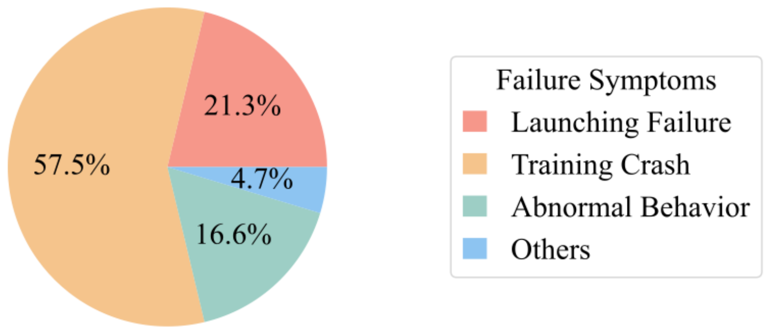
Launching Failure: 21.3%
- environment initialization
- model and dataset downloading

Training Crash: 57.5%
- iterative training failed

Abnormal Behavior: 16.6%
- training hanging
- training slowdown

74.1% during iterative training

Others: 4.7%
- resource unavailability



Classification of LLM training failure symptoms

Main Finding

Most LLM training failures (74.1%) occur during the iterative training stage, which can waste significant computational resources and training time.

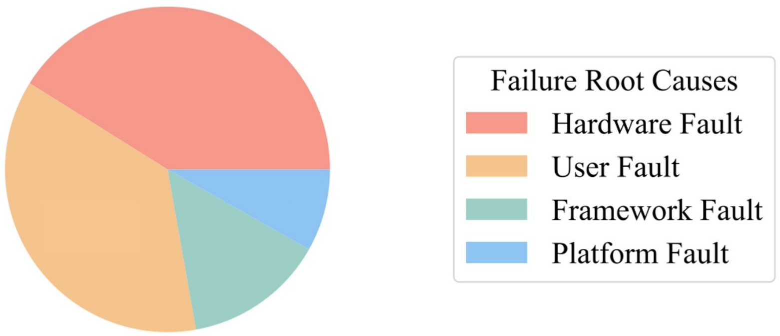
RQ2: LLM training failure root causes

1st : Hardware Fault:
- network fault
- accelerator fault
- node fault
- storage fault

2nd: User Fault
- configuration error
- program/script bug
- software incompatibility
- mis-operation

3rd : Framework Fault
- various LLM training frameworks and libraries:
  - PyTorch, DeepSpeed, MindSpore, etc.

4th : Platform Fault
- platform resource management
- platform job scheduling and configurations



Classification of LLM training failure root causes

Main Finding

LLM training failures stem from diverse causes, with hardware faults being significantly more prevalent than in traditional computing or DL workloads.

RQ3: LLM training failure diagnosis data

- **Manual diagnosis is challenging**

average diagnosis time: 34.7 hours | diagnosis > 24 hour: 41.9% | average training log size: 16.92GB

- **Monitor data for failure diagnosis**

Log-only Diagnosable:
- only training logs are involved

53.9%

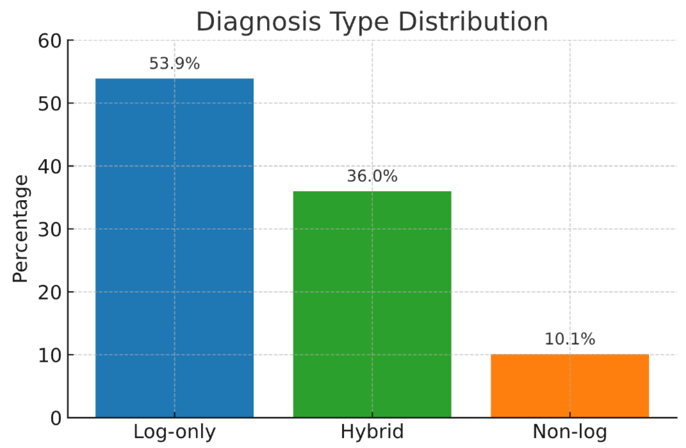
Hybrid Diagnosable:
- Both training logs and other monitoring data are jointly used

36.0%

89.9% failures

Non-log Diagnosable:
- Training logs do not provide clues

10.1%



Main Finding

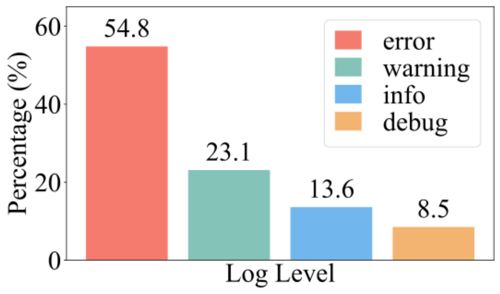
Training logs are crucial for diagnosing most (89.9%) LLM failures, but their sheer volume highlights the need for automated failure-indicating log identification.

Limitation for Existing Approaches

Goal: Automated identification of ***failure-indicating logs*** from substantial LLM training logs

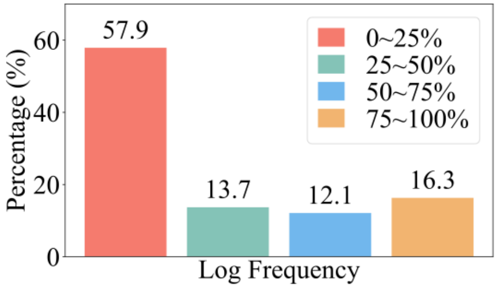
Features used in existing log-based anomaly detector:

Logging Level:
- **more serious logs (error)**



(a) Different logging levels.

Event Frequency:
- **infrequent log events**



(b) Different event frequencies.

Error Semantic:
- **logs with error semantics**

Our observations:
- Error-semantic logs from specific components/stages may not impact training.
- Successful training jobs often contain various error-semantic logs.
- Not all failure-indicating logs show explicit error semantics.

Main Finding

Existing Log AD methods struggle to effectively identify failure-indicating logs, as traditional anomalous log features are not suitable for LLM training log scenarios.

Distinct Patterns of LLM Training Logs

We have observed three distinct patterns of LLM training logs that can be used to *automatically pinpoint failure-indicating log*

① Cross-job Pattern

- In practice, each failed training job is usually associated with a series of successful jobs with identical settings (e.g., models and frameworks)
- The log events from both successful and failed jobs are typically noisy

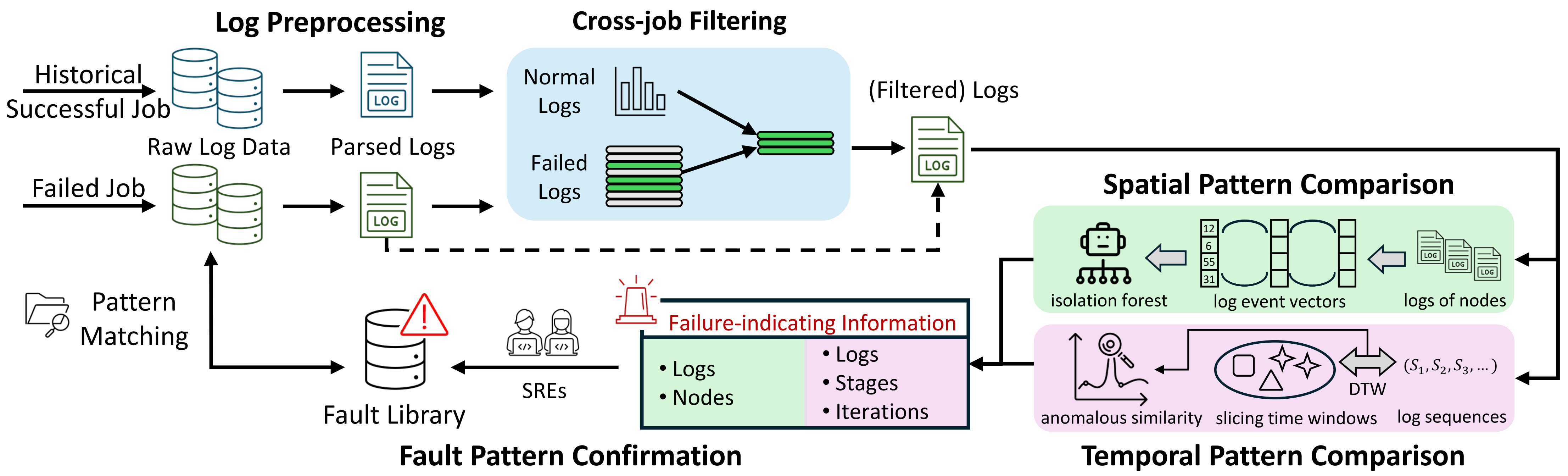
② Spatial Pattern

- Unlike traditional distributed systems, LLM training systems exhibit highly synchronized and nearly identical node workflows
- The distributed log sequences generated by different nodes are very similar

③ Temporal Pattern

- During the iterative training procedure, all nodes follow an identical workflow per iteration
- The deviation in log sequences across iterations can signal potential anomalies

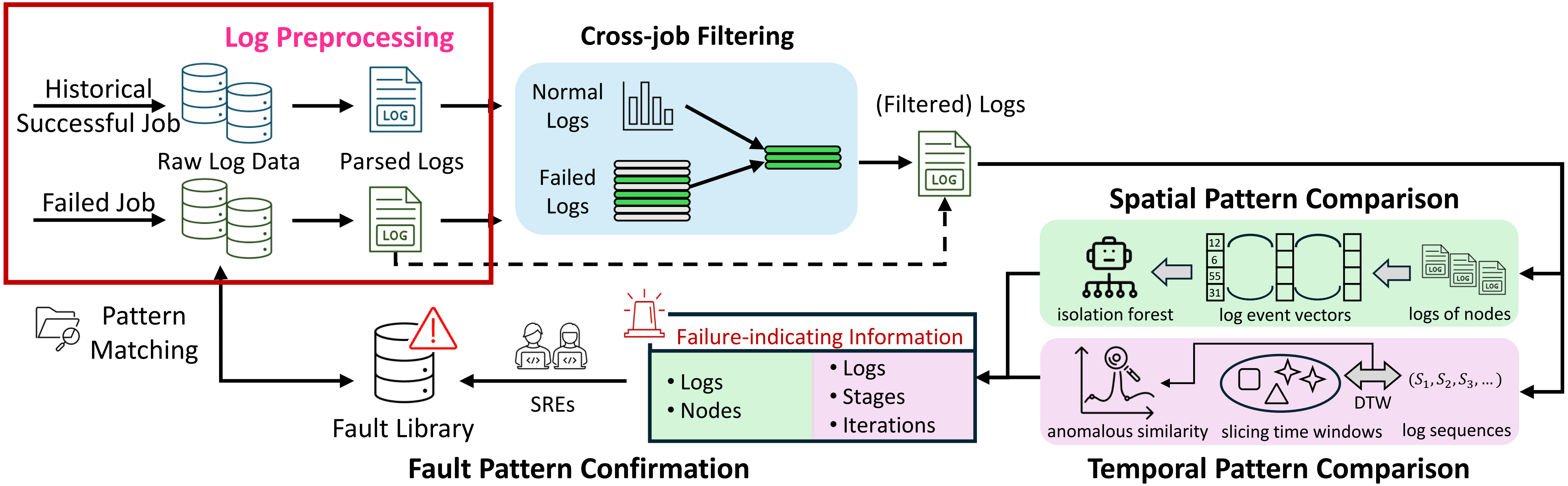
Our Framework: L4



L4: a Log-based Large-scale LLM training failure diagnosis framework

- automatically extract failure-indicating information from extensive training logs
- improve LLM training failure diagnostic efficiency

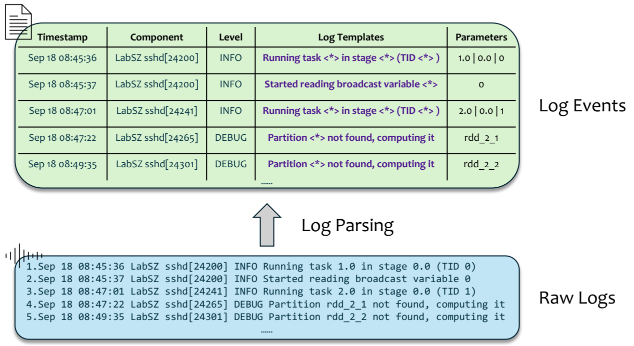
Our Framework: L4



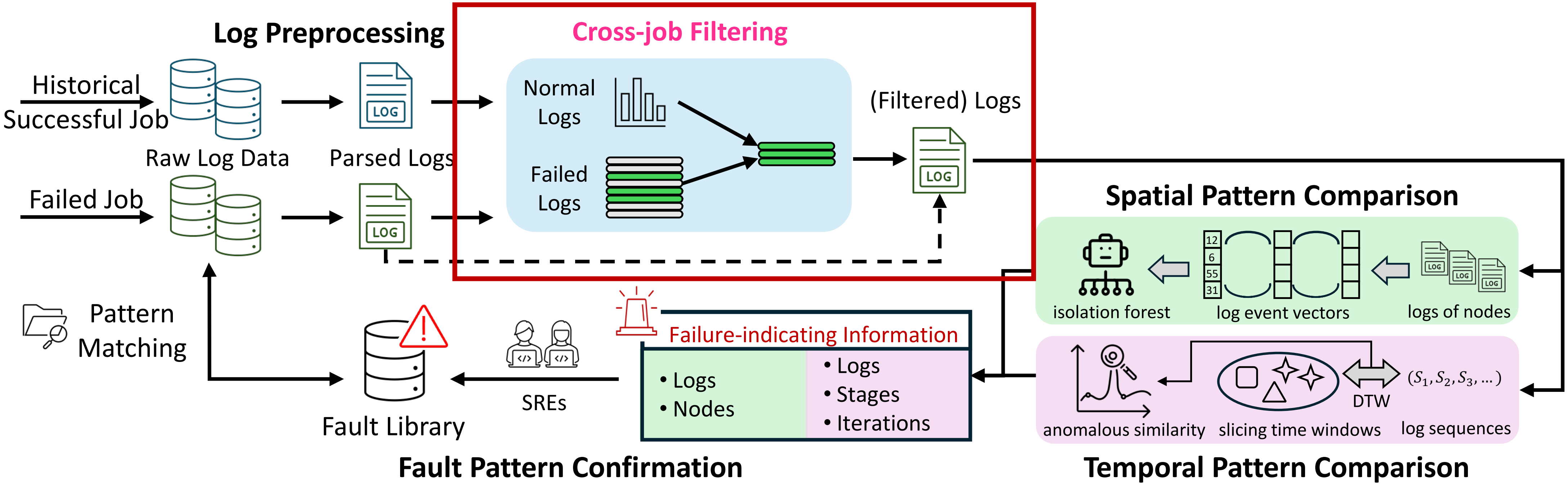
Log Preprocessing

Convert unstructured raw training logs into structured log events:

- adopt the widely-used and most efficient log parser, Drain



Our Framework: L4



Cross-job Filtering

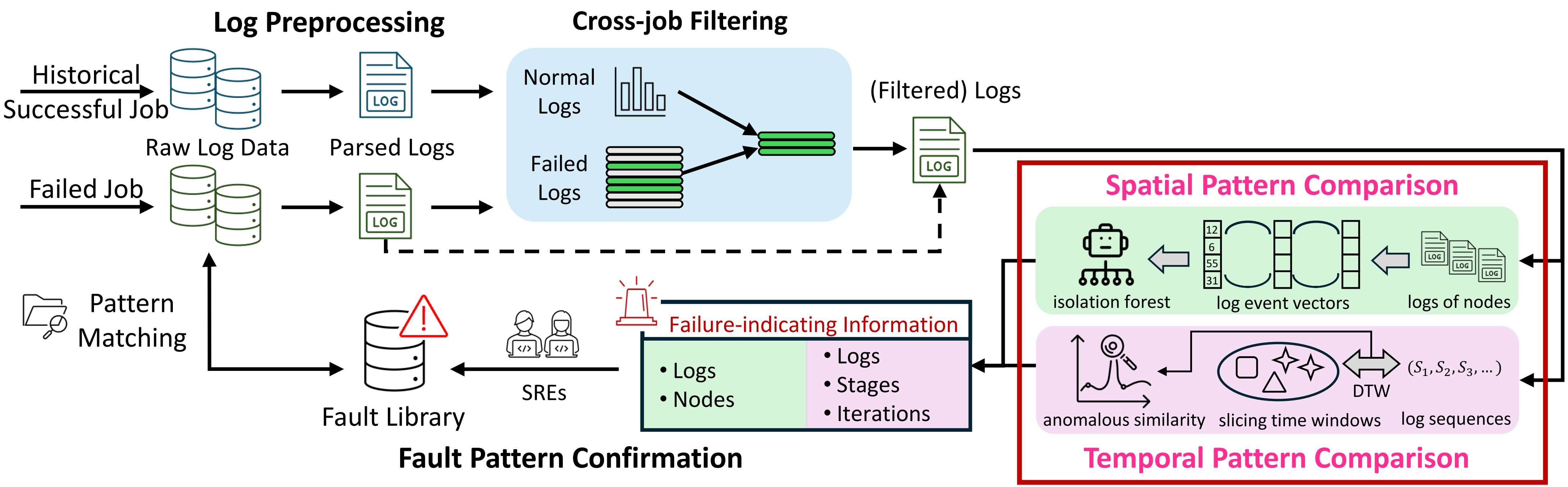
If there are related historical successful jobs:

- constructing a normal log event pool, denoted as $N = \{e_{n_1}, e_{n_2}, \cdots\}$
- filtering frequent log events within N

If there are no related historical successful jobs:

- using the parsed logs from failed jobs for analysis

Our Framework: L4



Spatial Pattern Comparision

Identify suspicious logs and nodes

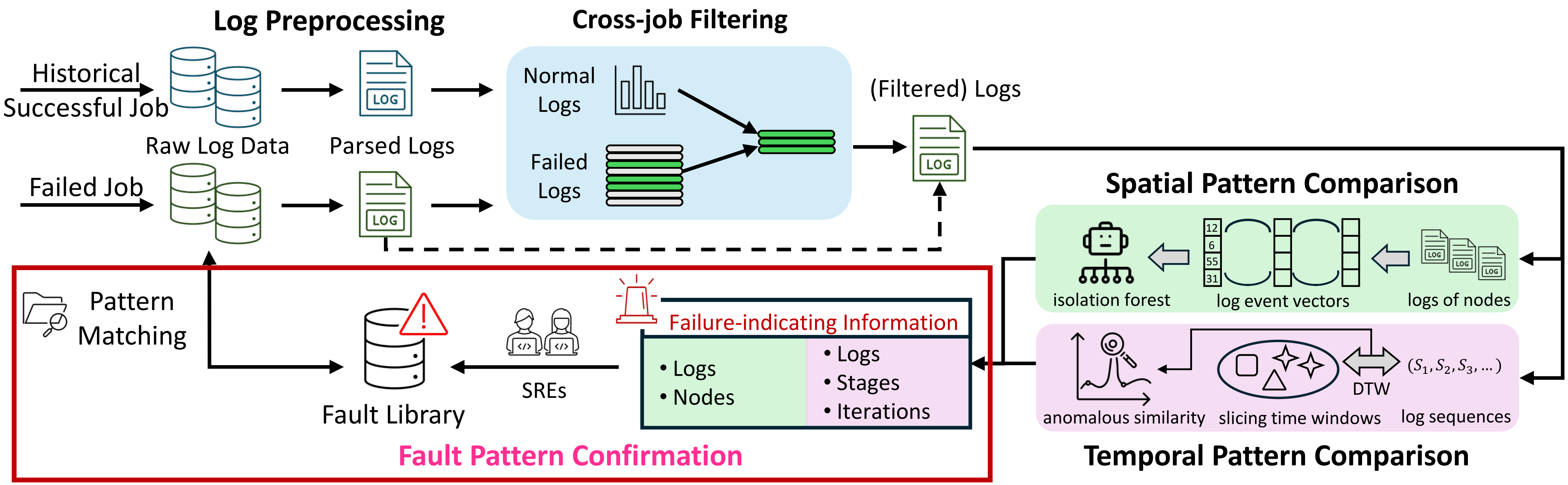
- trasforming the parsed log events of each node into log event vector V_i
- employing *Isolation Forest* to detect deviant log event vectors across these nodes
  - unsupervised and interpretability

Temporal Pattern Comparision

Identify suspicious logs and iterations

- converting the log events of each iteration into event sequence S_i
- employing dynamic time warping (DTW) distance for similarity measurement
- using slicing time window with 10 iterations for detect anomalous iterations

Our Framework: L4



Fault Pattern Confirmation

Summarized fault patterns:

- failure-indicating logs, nodes, iterations and stages
- manually confirmed by engineers for future matching
- improving the efficiency of handling similar failures in the future

Evaluation

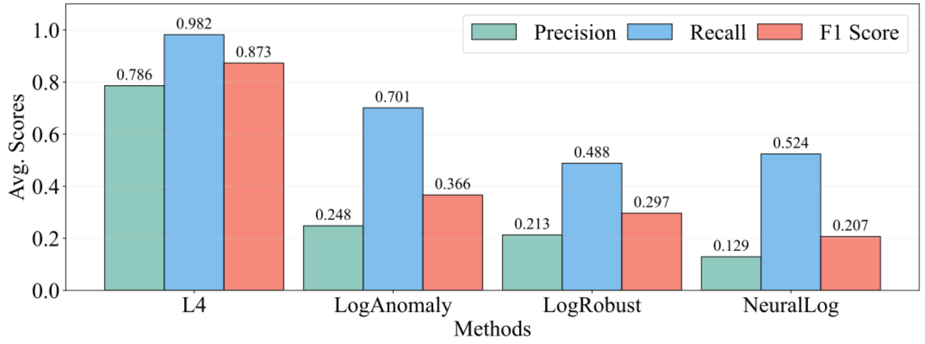
Research Questions

- RQ4: How effective is L4 in identifying failure-indicating logs
- RQ5: How effective is L4 in locating faulty nodes?

Evaluation Datasets

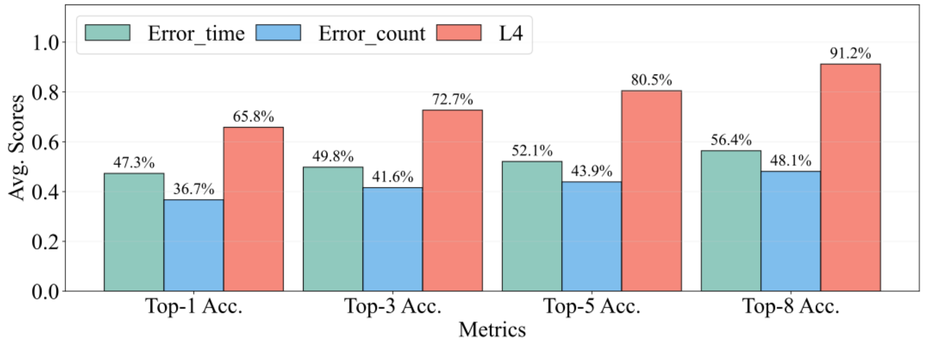
- 100 randomly sampled failed LLM training jobs
  - averaging 632 accelerators and 12.3GB of training logs
- 42 cases were caused by hardware faults related to specific nodes

RQ4: failure-indicating log identification



L4 achieved the best accuracy in failure-indicating log identification, with an F1 score of 0.873.

RQ5: Faulty node localization



L4 achieved the best accuracy in faulty node localization, with a Top-8 Accuracy of 91.2.

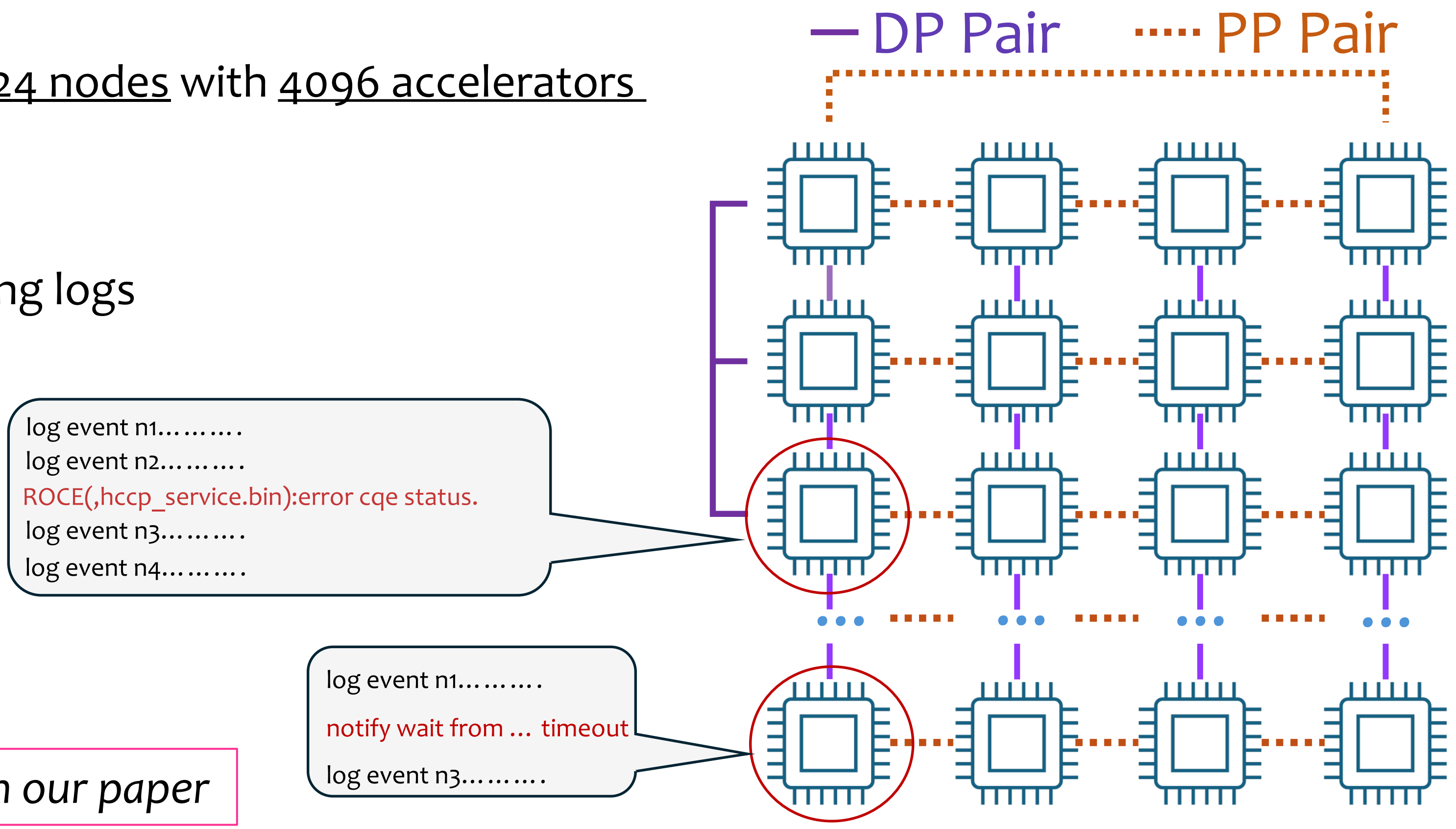
Case Study

fine-grained localization of hardware fault

deployed for one year (since Jun. 2024)

an LLM training job involving 1024 nodes with 4096 accelerators

failed after 20 hours of training,
producing 71 GBs of latest training logs



More case studies can be found in our paper

Discussion

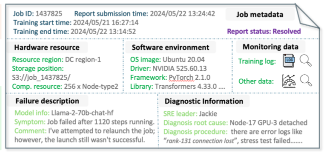
- **Future Research Directions**
  - LLM Training Monitoring
    - The observability of LLM training framework is still evolving
      - the logging quality should be improved
      - not only logs, but also efficient and effective profiling approach
  - Multi-modal Failure Diagnosis
    - *36.0% of failures require hybrid monitoring data for accurate diagnosis*
    - Integrating multi-modal monitoring data for automated failure diagnosis and root cause analysis is essential for LLM training reliability

Conclusion

Empirical Study on LLM Training Failures

Study Platform

Our multi-tenant production LLM training Platform-X supporting hundreds of internal users and partner companies



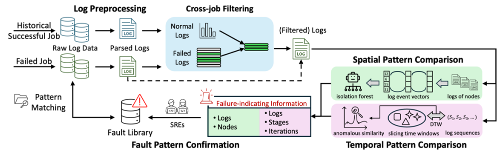
Study Subject

Collected 428 failure reports of failed large-scale LLM training jobs in Platform-X from May 2023 to April 2024
- average model size: 72.8B parameters
- average accelerators: 941 GPUs or NPUs

Study RQs
- RQ1: What are the common symptoms of LLM training failures?
- RQ2: What are the common root causes of LLM training failures?
- RQ3: What monitor data sources are typically used to diagnose LLM training failures?

Empirical Study

Our Framework: L4



L4: a Log-based Large-scale LLM training failure diagnosis framework

- automatically extract failure-indicating information from extensive training logs
- improve LLM training failure diagnostic efficiency

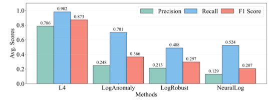
Approach

Evaluation

Research Questions
- RQ4: How effective is L4 in identifying failure-indicating logs
- RQ5: How effective is L4 in locating faulty nodes?

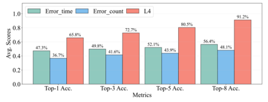
Evaluation Datasets
- 100 randomly sampled failed LLM training jobs
  - averaging 632 accelerators and 12.3GB of training logs
- 42 cases were caused by hardware faults related to specific nodes

RQ4: failure-indicating log identification



L4 achieved the best accuracy in failure-indicating log identification, with an F1 score of 0.873.

RQ5: Faulty node localization



L4 achieved the best accuracy in faulty node localization, with a Top-8 Accuracy of 91.2.

Experiment

Pre-print paper

LogPAI website

Thank you for listening

Q & A

Contact: zhjiang@link.cuhk.edu.hk