



# A Large-scale Evaluation for Log Parsing Techniques: How Far Are We?

**Zhihan Jiang**<sup>1</sup>, Jinyang Liu<sup>1</sup>, Junjie Huang<sup>1</sup>, Yichen Li<sup>1</sup>, Yintong Huo<sup>1</sup>,  
Jiazhen Gu<sup>1</sup>, Zhuangbin Chen<sup>2</sup>, Jieming Zhu and Michael R. Lyu<sup>1</sup>

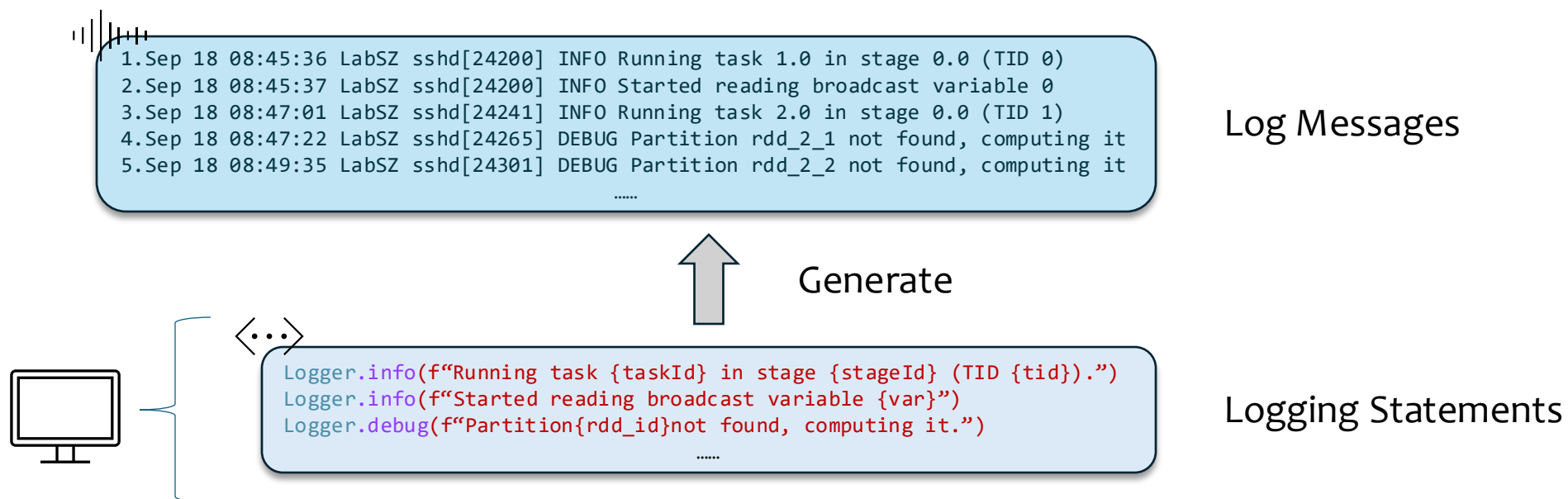
<sup>1</sup>The Chinese University of Hong Kong,

<sup>2</sup>Sun Yat-sen University



# System Log

**Logs** are generated by logging statements and record runtime information of software systems.



# System Log

Automated Log Analysis is important for:



Anomaly Detection



Fault Localization



Software Diagnosis



```
1.Sep 18 08:45:36 LabSZ sshd[24200] INFO Running task 1.0 in stage 0.0 (TID 0)
2.Sep 18 08:45:37 LabSZ sshd[24200] INFO Started reading broadcast variable 0
3.Sep 18 08:47:01 LabSZ sshd[24241] INFO Running task 2.0 in stage 0.0 (TID 1)
4.Sep 18 08:47:22 LabSZ sshd[24265] DEBUG Partition rdd_2_1 not found, computing it
5.Sep 18 08:49:35 LabSZ sshd[24301] DEBUG Partition rdd_2_2 not found, computing it
.....
```

Log Messages



Generate



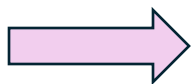
```
Logger.info(f"Running task {taskId} in stage {stageId} (TID {tid}).")
Logger.info(f"Started reading broadcast variable {var}")
Logger.debug(f"Partition{rdd_id}not found, computing it.")
.....
```

Logging Statements



# Log Parsing

Logs are unstructured, which are not suitable for *understanding, analyzing* and *storing*.



## Log Parsing

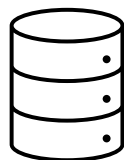
Converting raw logs into structured formats is a critical prerequisite step!

```
Sep 18 08:47:22 DEBUG Partition rdd_2_1 not found in 127.0.0.1
Sep 18 08:47:35 DEBUG Partition rdd_2_2 not found in 127.0.0.1
Sep 18 08:47:49 DEBUG Partition rdd_2_1 not found in 127.0.0.1
Sep 18 08:48:17 DEBUG Partition rdd_2_3 not found in 127.0.0.1
```



Timestamp	Event	Level	Log Templates	Parameters
Sep 18 08:47:22	e1	DEBUG	Partition <*> not found in <*>	rdd_2_1   127.0.0.1
Sep 18 08:47:35	e1	DEBUG	Partition <*> not found in <*>	rdd_2_2   127.0.0.1
Sep 18 08:47:49	e1	DEBUG	Partition <*> not found in <*>	rdd_2_1   127.0.0.1
Sep 18 08:48:17	e1	DEBUG	Partition <*> not found in <*>	rdd_2_3   127.0.0.1

Substantial Raw Log



How to understand and process these complex logs?



Structured Parsed Log



Log Analysis




Four same events happened on **three different partitions** (*rdd\_2\_1, rdd\_2\_2, rdd\_2\_3*) and **the same address** (*127.0.0.1*)



# Log Parsing

**Goal:** to distinguish between constant part and variable part.




Timestamp	Component	Level	Log Templates	Parameters
Sep 18 08:45:36	LabSZ sshd[24200]	INFO	Running task <*> in stage <*> (TID <*> )	1.0   0.0   0
Sep 18 08:45:37	LabSZ sshd[24200]	INFO	Started reading broadcast variable <*>	0
Sep 18 08:47:01	LabSZ sshd[24241]	INFO	Running task <*> in stage <*> (TID <*> )	2.0   0.0   1
Sep 18 08:47:22	LabSZ sshd[24265]	DEBUG	Partition <*> not found, computing it	rdd_2_1
Sep 18 08:49:35	LabSZ sshd[24301]	DEBUG	Partition <*> not found, computing it	rdd_2_2
.....				

Parsed Results



Log Parsing



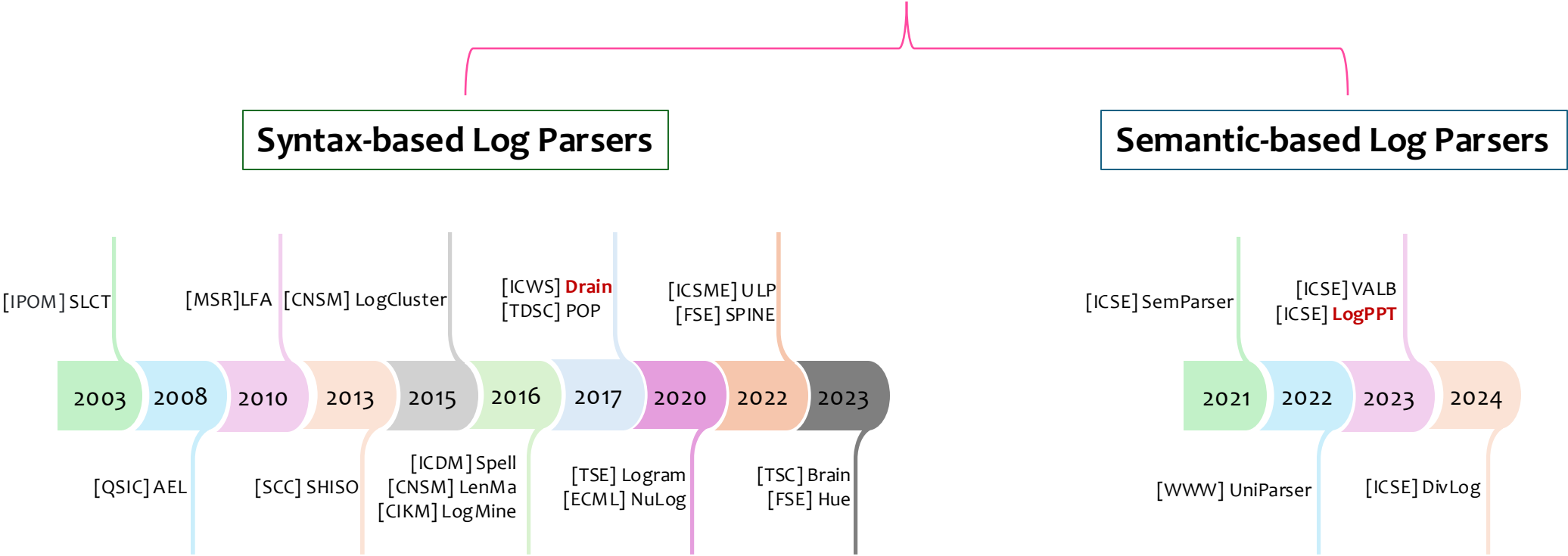
```
1.Sep 18 08:45:36 LabSZ sshd[24200] INFO Running task 1.0 in stage 0.0 (TID 0)
2.Sep 18 08:45:37 LabSZ sshd[24200] INFO Started reading broadcast variable 0
3.Sep 18 08:47:01 LabSZ sshd[24241] INFO Running task 2.0 in stage 0.0 (TID 1)
4.Sep 18 08:47:22 LabSZ sshd[24265] DEBUG Partition rdd_2_1 not found, computing it
5.Sep 18 08:49:35 LabSZ sshd[24301] DEBUG Partition rdd_2_2 not found, computing it
.....
```

Raw Logs



# Existing Log Parsers

A wide range of **data-driven** methods have been proposed to parse logs.



**Evaluating these diverse log parsers is crucial!**



# Current Benchmark



Loghub-2k Datasets: randomly sampled 2,000 log lines from various systems

logparser Public

Watch 60

Fork 550

Star 1.5k

Dataset	Description	Time Span	Data Size	#Messages	#Templates (total)	#Templates (2k)
Distributed system logs						
HDFS	Hadoop distributed file system log	38.7 hours	1.47 GB	11,175,629	30	14
Hadoop	Hadoop mapreduce job log	N.A.	48.61 MB	394,308	298	114
Spark	Spark job log	N.A.	2.75 GB	33,236,604	456	36
ZooKeeper	ZooKeeper service log	26.7 days	9.95 MB	74,380	95	50
OpenStack	OpenStack software log	N.A.	60.01 MB	207,820	51	43
Supercomputer logs						
BGL	Blue Gene/L supercomputer log	214.7 days	708.76 MB	4,747,963	619	120
HPC	High performance cluster log	N.A.	32.00 MB	433,489	104	46
Thunderbird	Thunderbird supercomputer log	244 days	29.60 GB	211,212,192	4,040	149
Operating system logs						
Windows	Windows event log	226.7 days	26.09 GB	114,608,388	4,833	50
Linux	Linux system log	263.9 days	2.25 MB	25,567	488	118
Mac	Mac OS log	7.0 days	16.09 MB	117,283	2,214	341
Mobile system logs						
Android	Android framework log	N.A.	3.38 GB	30,348,042	76,923	166
HealthApp	Health app log	10.5 days	22.44 MB	253,395	220	75
Server application logs						
Apache	Apache server error log	263.9 days	4.90 MB	56,481	44	6
OpenSSH	OpenSSH server log	28.4 days	70.02 MB	655,146	62	27
Standalone software logs						
Proxifier	Proxifier software log	N.A.	2.42 MB	21,329	9	8

47K  
VIEWS

70K  
DOWNLOADS

Show less details

	All versions	This version
Views	47,485	5,537
Downloads	69,537	14,199
Data volume	38.2 TB	9.9 TB

"Loghub: A Large Collection of System Log Datasets for AI-driven Log Analytics ", ISSRE'23

"Guidelines for assessing the accuracy of log message template identification techniques", ICSE'22



# Existing Log Parsers on Loghub-2k

Existing log parsers achieved encouraging performance on Loghub-2k

Syntax-based Log Parsers

Semantic-based Log Parsers

**The performance of log parsers in real-world systems remains unsatisfactory, especially when log data are diverse and complex.**



"Guidelines for assessing the accuracy of log message template identification techniques", ICSE'22

"An empirical study of log analysis at Microsoft", FSE'22

"Investigating and improving log parsing in practice", FSE'22





# Limitation of Current Benchmark

---

## It is small in scale, affecting the representativeness.

- Loghub-2k may not be able to reflect the diverse and complex characteristics of log data observed in production environments.
- The performance of data-driven log parsers could be affected by the limited scale of Loghub-2k and may not generalize well to real-world scenarios with much larger and diverse log data.

## It lacks comprehensive evaluation metrics.

- Most of existing studies only employ message-level metrics, which tend to favor frequently occurring log templates
- Real-world logs are highly imbalanced, with frequent log templates (e.g., heartbeat logs) dominating the overall performance and potentially masking errors in the processing of infrequent templates.

## It only reports the performance on entire datasets.

- The overall performance lacks a fine-grained analysis of accuracy for logs with different characteristics.
- We have identified two critical types of logs that play an essential role in production software analysis:
  - **Infrequent log templates:** represent rare system events that require particular attention
  - **Parameter-intensive log templates:** provide informative details about system status and associated events



# A new benchmark for log parsers

---

**Goal: evaluate log parsers in a more rigorous and practical setting**

## Our Work:

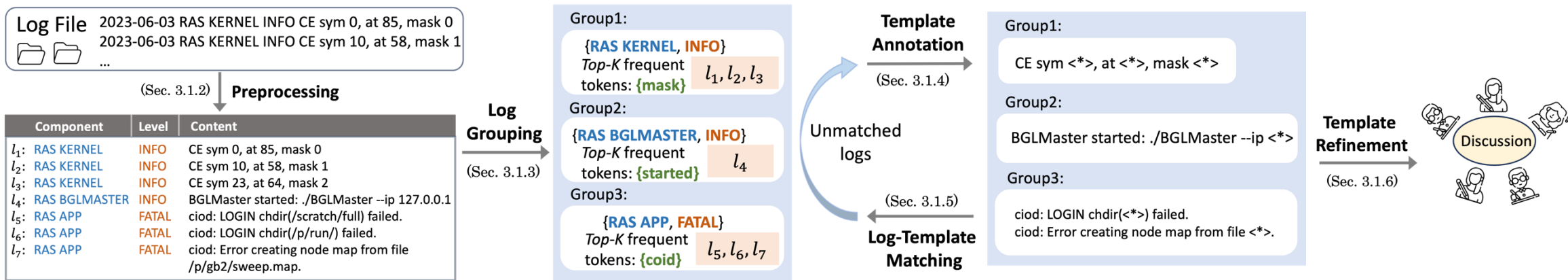
- ❖ A new version of large-scale annotated log datasets for log parsing, i.e., **Loghub-2.0**
  - 14 datasets from various software systems, each containing **3.6 million** log messages on average.
- ❖ A more comprehensive benchmarking protocol to evaluate existing log parsers
  - template-level metric (FGA) to mitigate the sensitivity of message-level metrics (GA) to imbalanced log data
  - investigation of the performance on log templates with different frequencies and parameter counts
- ❖ A comprehensive re-evaluation of 13 syntax-based and 2 semantic-based log parsers
  - more practical guidance for researchers and practitioners on understanding the characteristics of these log parsers



# Dataset Construction

Problem: How can we efficiently annotate millions of logs from various systems?

## Our proposed annotation framework





# Dataset Construction

## Our annotated results: Loghub-2.0

System	Dataset	# Templates (Loghub-2k)	# Templates (Loghub-2.0)	# Annotated Logs (Loghub-2.0)
Distributed systems	Hadoop	114	236	179,993
	HDFS	14	46	11,167,740
	OpenStack	43	48	207,632
	Spark	36	236	16,075,117
	Zookeeper	50	89	74,273
Super- computer systems	BGL	120	320	4,631,261
	HPC	46	74	429,987
	Thunderbird	149	1,241	16,601,745
Operating systems	Linux	118	338	23,921
	Mac	341	626	100,314
Server application	Apache	6	29	51,977
	OpenSSH	27	38	638,946
Standalone software	HealthApp	75	156	212,394
	Proxifier	8	11	21,320
Average		81.9	249.1	3,601,187



# Study Design

---

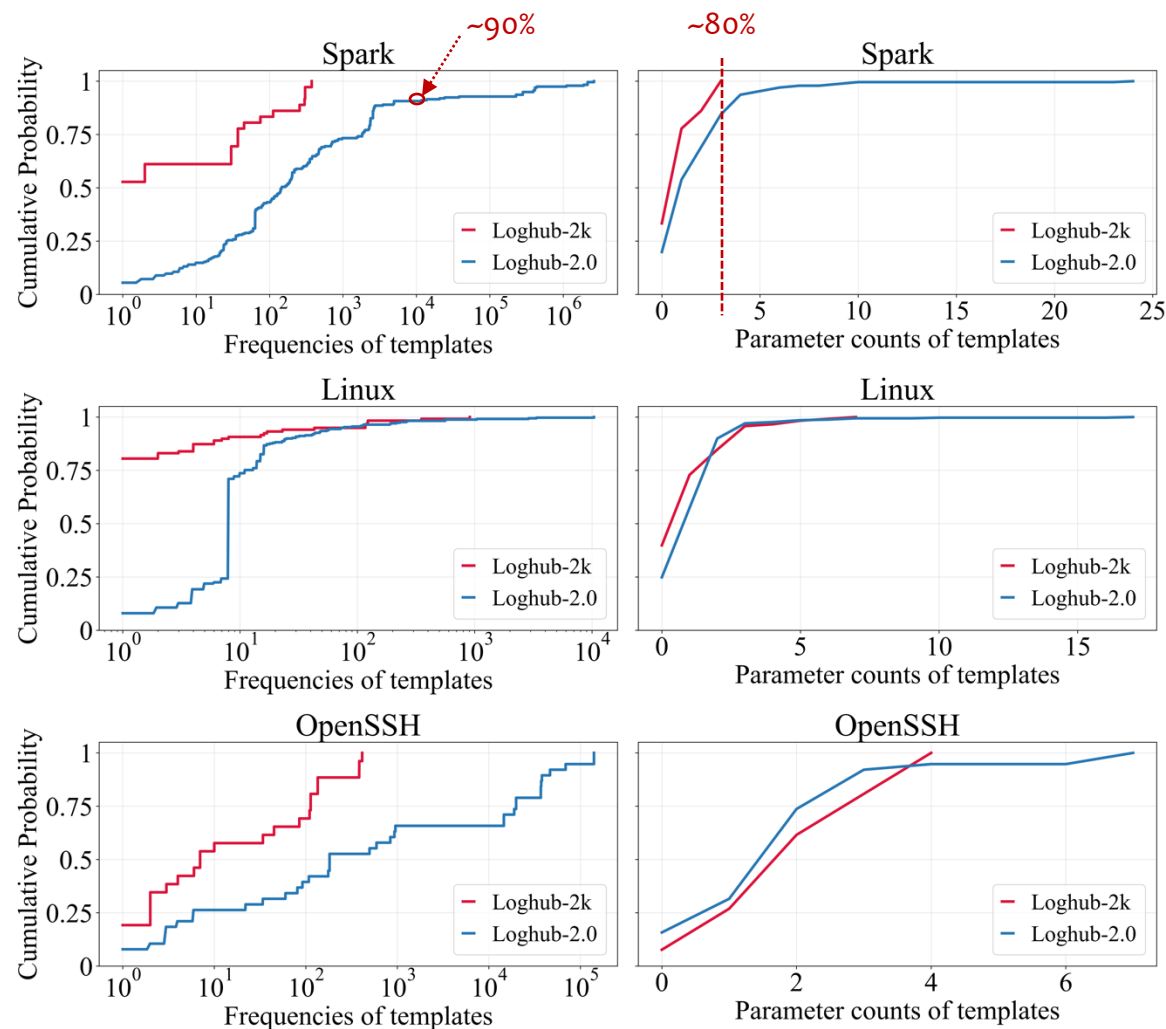
We focus on the following three main Research Questions (RQs):

**RQ1: What are the differences between Loghub-2.0 and Loghub-2k?**

**RQ2: How does the performance of log parsers differ when applied to Loghub-2.0 compared to Loghub-2k?**

**RQ3: What is the performance of log parsers on logs with varying characteristics?**

# RQ1: Differences between Loghub-2.0 and Loghub-2k

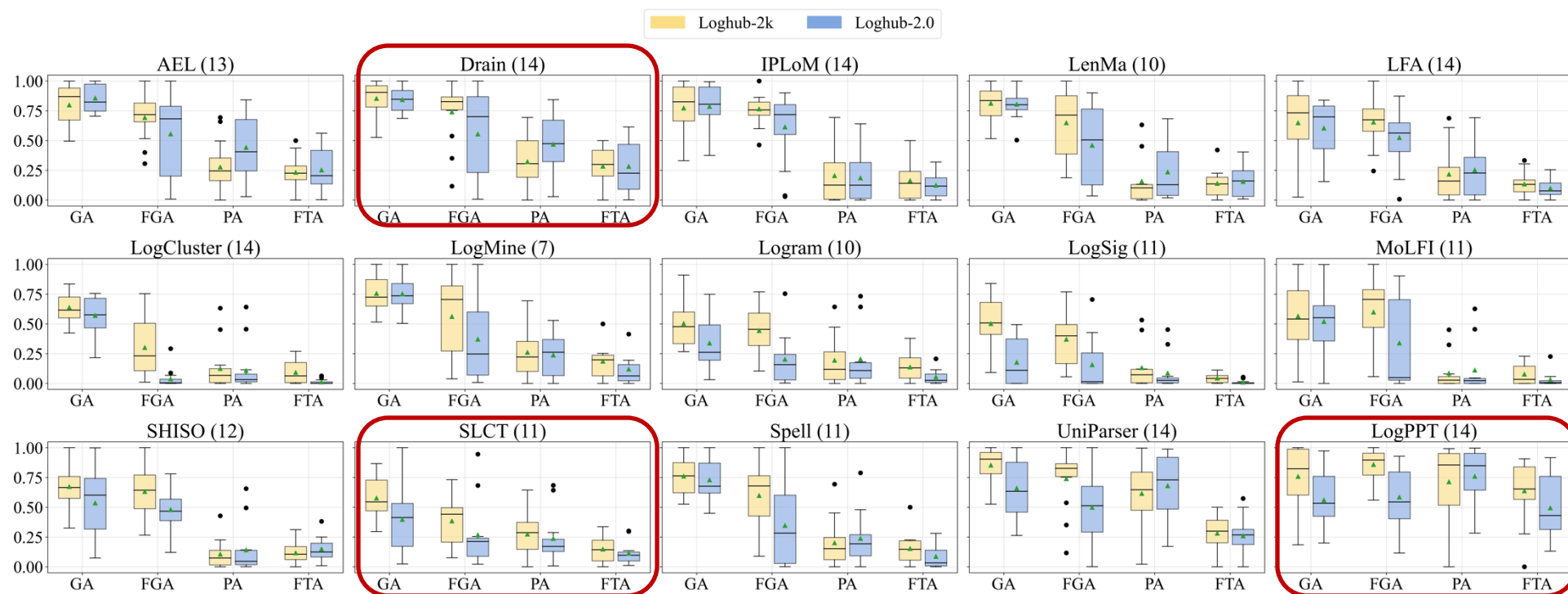


## Finding 1.

The distributions of **template frequencies** and **parameter counts** exhibit significant differences.

- Loghub-2.0 exhibits **a more pronounced imbalance** in template frequencies
- Loghub-2.0 contains more log templates, and has **a larger parameter count** on average

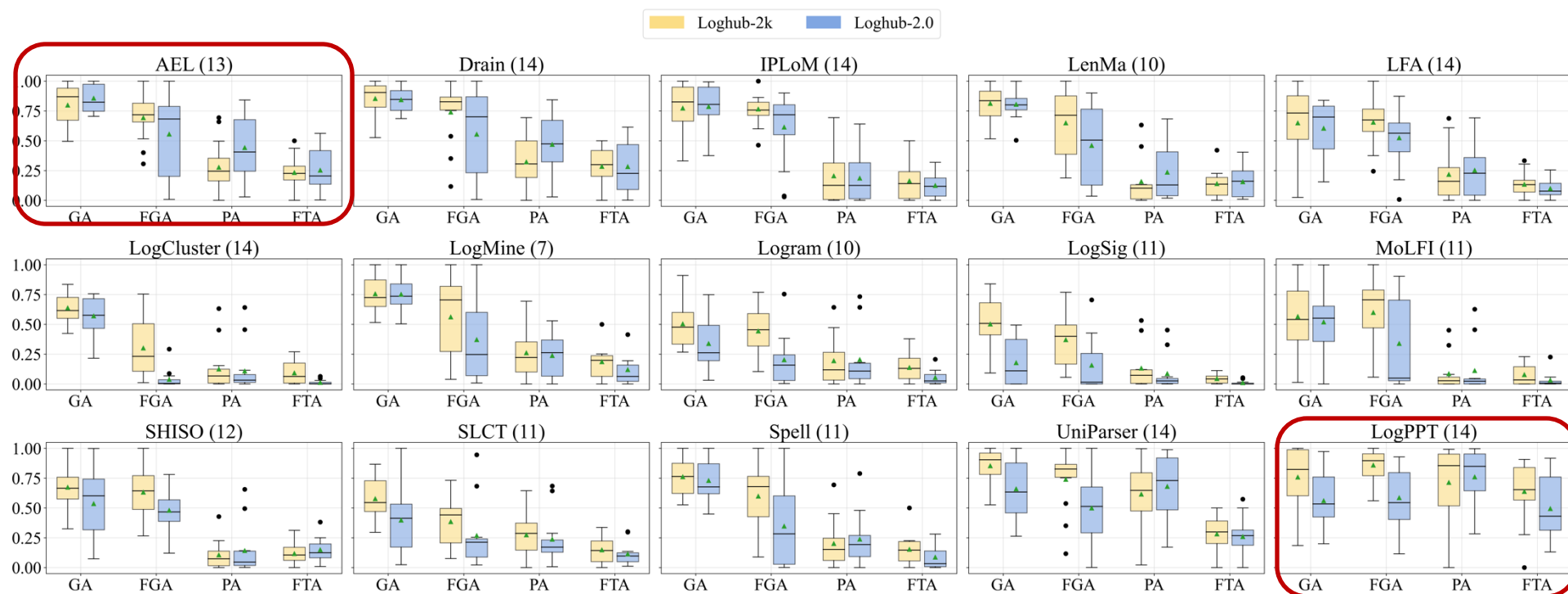
# RQ2: Performances on Loghub-2.0 and Loghub-2k



## Finding 2.

On Loghub-2.0, existing log parsers experience **a performance drop** and **an increase in variance** across all metrics.

# RQ2: Performances on Loghub-2.0 and Loghub-2k



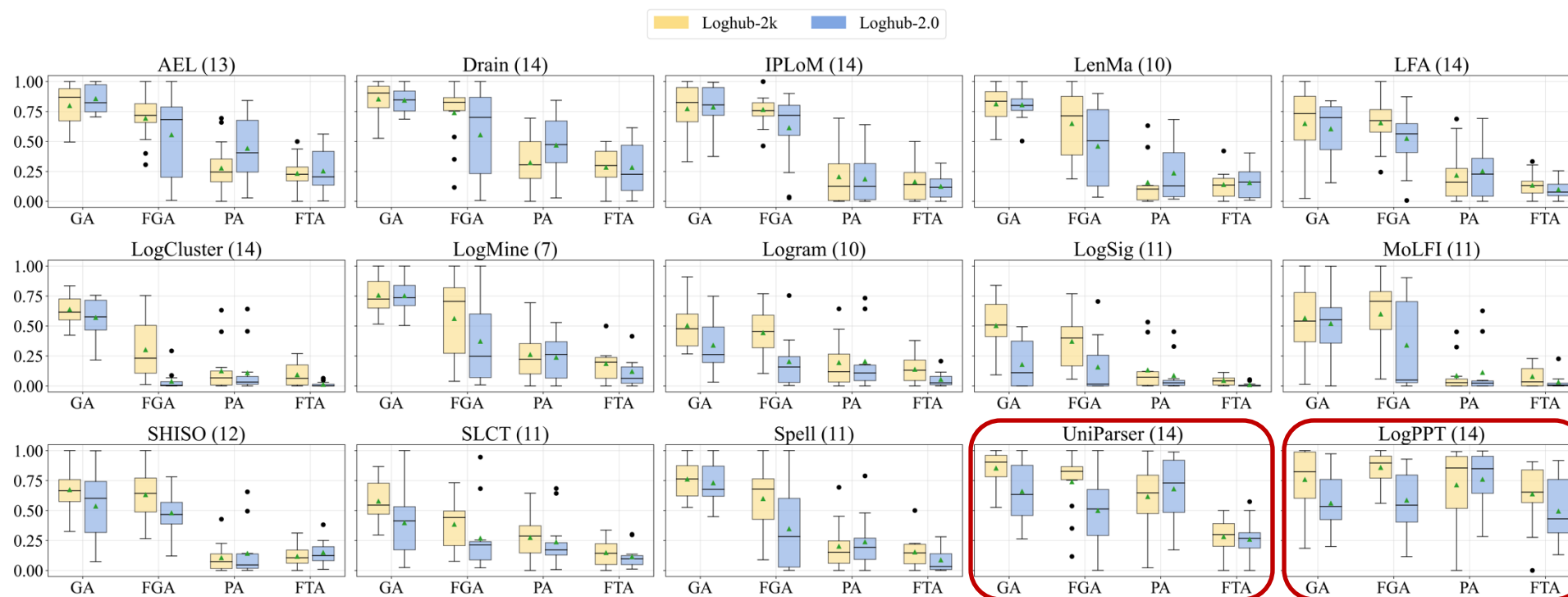
## Finding 3.

Message-level metrics (GA and PA) generally yield higher results than template-level metrics (FGA and FTA) due to their sensitivity to imbalanced log data.

These differences are more pronounced in highly imbalanced Loghub-2.0 datasets.



# RQ2: Performances on Loghub-2.0 and Loghub-2k

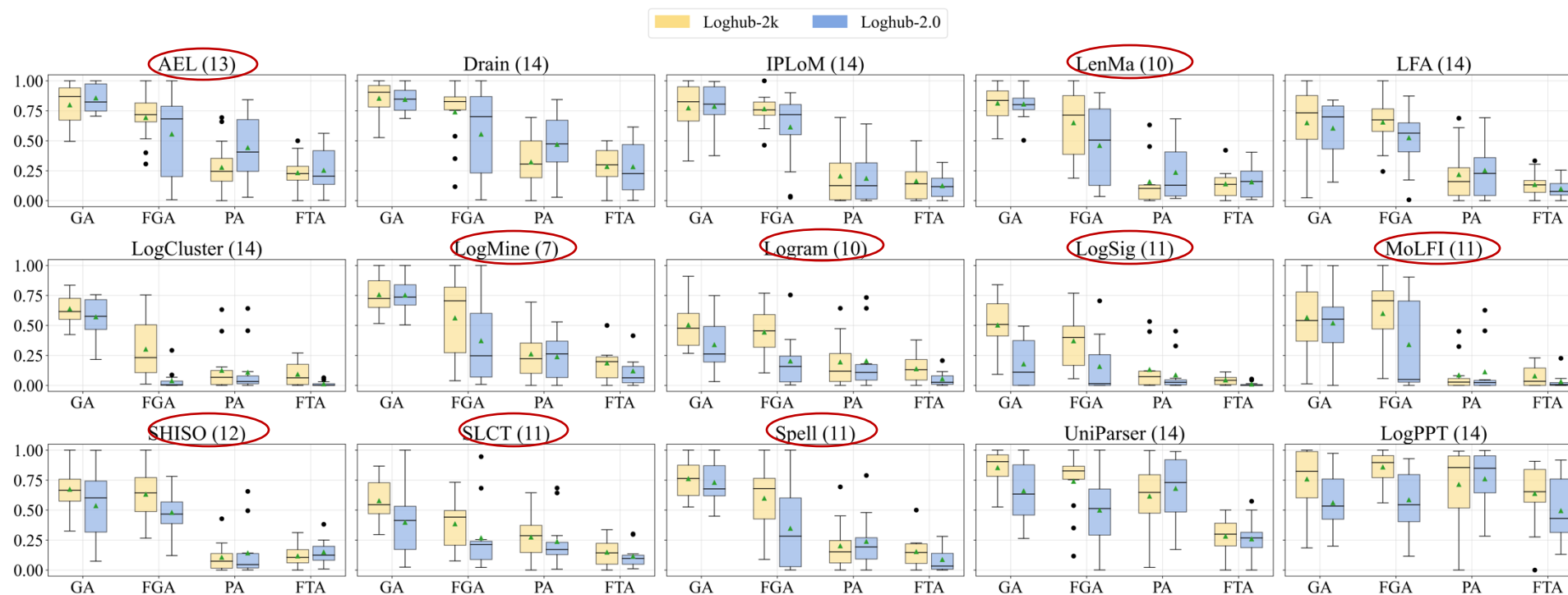


## Finding 4.

Semantic-based log parsers excel at parsing individual logs, as shown by their higher PA. However, they perform poorly on grouping-related metrics due to ignoring global information.

Moreover, their performance decline on larger, more diverse datasets, especially with limited labeled training log data.

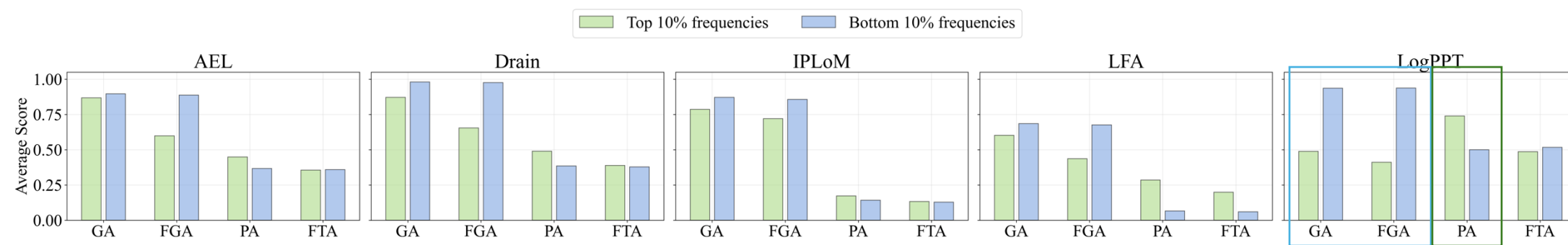
# RQ2: Performances on Loghub-2.0 and Loghub-2k



## Finding 5.

9 out of 15 log parsers are unable to process all 14 datasets of Loghub-2.0 within a reasonable 12-hour timeframe.

# RQ3: Performances on logs with different characteristics



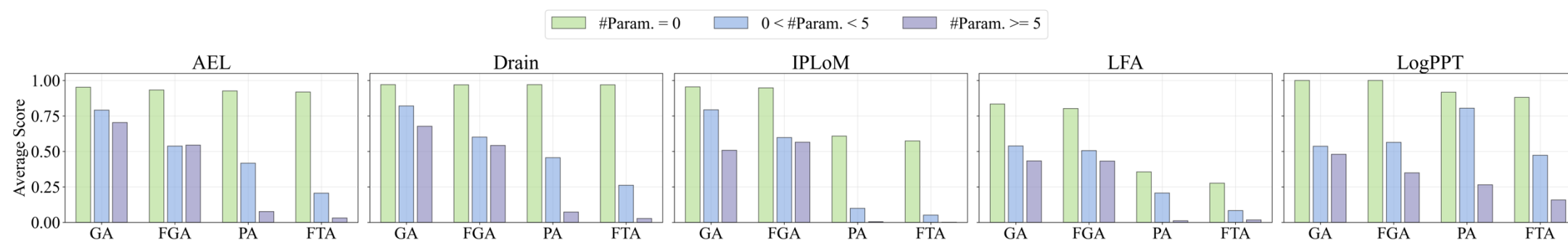
Existing log parsers show varying effectiveness with templates of different frequencies.



## Finding 6.

- **Lower grouping-related metrics (GA and FGA) for frequent templates**, as the grouping is more challenging for templates with more log messages.
- **Lower parsing-related metrics (PA and FTA) for infrequent templates**, as less evidence (e.g., training data) is available to guide the accurate parsing of log messages.

# RQ3: Performances on logs with different characteristics



## Finding 7.

Despite high scores on entire datasets, log parsers still struggle with parameter-intensive log templates.



# Summary

---

- Loghub-2.0 exhibits significantly different characteristics compared to the commonly used Loghub-2k.
- Drain is the most effective parsers for grouping log messages.
- Semantic-based methods (e.g., LogPPT) excel in accurately parsing individual logs.
- Despite the encouraging results shown in the Loghub-2k, the parsing performance remains unsatisfactory when applied to Loghub-2.0, especially for infrequent and parameter-intensive logs.
- The efficiency of most log parsers fails to meet the demands of real-world application scenarios.



# Implications

---

**Consider both levels of metrics in combination.**

**Evaluate the performance across logs with different characteristics.**

**Try to combine semantic and statistical information.**

**Place greater emphasis on parsing efficiency.**

# Conclusion

## Limitation of Current Benchmark

### It is small in scale, affecting the representativeness.

- Loghub-2k may not be able to reflect the diverse and complex characteristics of log data observed in production environments.
- The performance of data-driven log parsers could be affected by the limited scale of Loghub-2k and may not generalize well to real-world scenarios with much larger and diverse log datasets.

### It lacks comprehensive evaluation metrics.

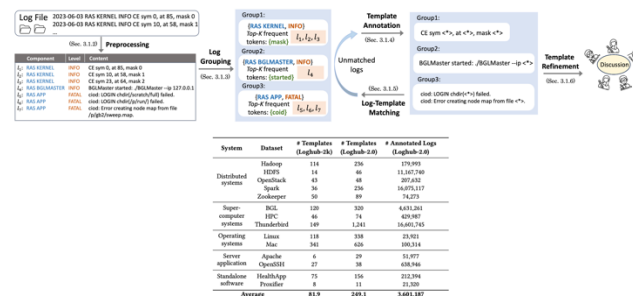
- Most of existing studies only employ message-level metrics, which tend to favor frequently occurring log templates.
- Real-world logs are highly imbalanced, with frequent log events (e.g., heartbeat logs) dominating the overall performance and potentially masking errors in the processing of infrequent templates.

### It only reports the performance on entire datasets.

- The overall performance lacks a fine-grained analysis of accuracy for logs with different characteristics.
- We have identified two critical types of logs that play an essential role in production system maintenance:
  - Infrequent log templates:** represent rare system events that require particular attention
  - Parameter-intensive log templates:** provide informative details about system status and associated entities

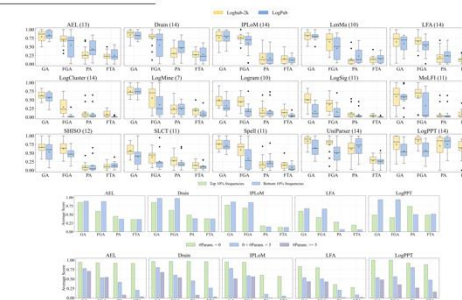
## Limitation of Loghub-2k

## Construction of Loghub-2.0



## Our Loghub-2.0

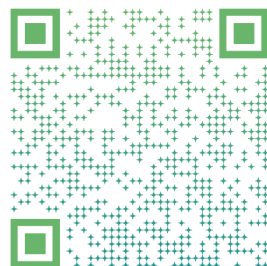
## Re-evaluation of 15 existing log parsers



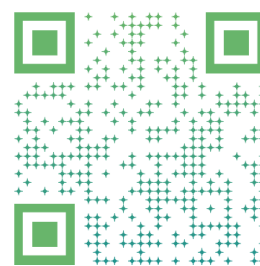
## Study Results



Pre-print paper



Project



LogPAI website

Thank you for listening

Q & A

Contact: [zhjiang@link.cuhk.edu.hk](mailto:zhjiang@link.cuhk.edu.hk)