SLS-Bench: A Benchmark for Incident Log Summarization with Synthetic Observability Data

Valerii Iakovlev Markus Spanring Martin Flechl

Can an LLM agent analyze the logs of a failing system and write a comprehensive incident report, like an SRE would?

SLS-Bench in a nutshell

- Engineers diagnose incidents by searching large log streams and writing up what happened. LLM agents could help engineers to write these reports.
- No benchmark tests this end to end. Existing ones use short log snippets, narrow tasks or keyword answers, because real incident logs with comprehensive incident reports are hard to collect.
- SLS-Bench has 90 tasks. Each contains a large synthetic log file and a reference report, both grounded in a unique real incident report.
- We test 14 LLMs. Frontier models produce useful reports but at high cost, and open-weight models lag behind. Models find the core fault chain well, but might struggle to provide complete evidence and reasoning to support it.

Benchmark tasks

Each task is derived from a unique real-world incident report, and consists of a question, log file, and reference answer.

- (a) Each task has the same question asking an agent to analyze the log file and provide a structured incident report.
- (a) Each task contains one log file (56k lines on average) and a reference answer (2400 words on average).
- (b) Prediction and evaluation pipelines. An agent receives the question and a log file. It then uses Python to analyze the logs and provides a response. The response is graded against the reference by an LLM judge on the 0 (worst) to 1 (best) scale.

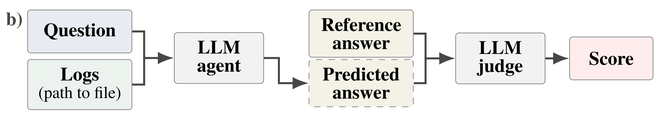
a)

Question
Analyze the logs and write an incident summary. The answer must contain system description, timeline, and diagnosis...

Logs (CSV)
```
ts,level,service,message,host,trace_id
12:01:03,INFO,nginx,request_start...
12:01:04,WARN,postgres,lock_wait...
12:01:04,ERROR,api,timeout...
... (20k-100k rows; 56k on average)
```

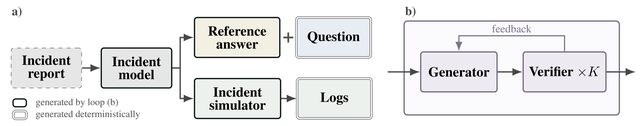
Reference answer
Reconstructed system
<system purpose, main components and interactions>
Timeline
<baseline, trigger, progression phases; with summary, evidence, reasoning, and certainty per phase>
Diagnosis
<main fault chain, evidence, and certainty>

b)

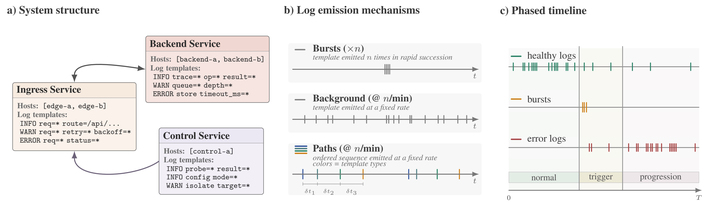


From incident reports to benchmark tasks

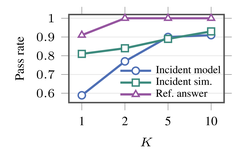
- We collected 343 public incident reports and, after de-duplication and quality filtering, kept the 90 most informative ones.
- Each report is converted into a structured incident model, which is then used to generate a benchmark task (question, logs, and reference answer) via the generator–verifier loop.



- Incident model enables inferring missing information from diverse incident reports and converting them into a structured representation in a consistent, controllable, and verifiable way.

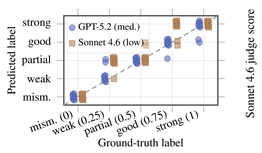


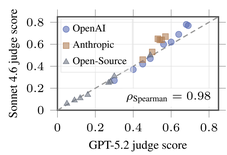
During generation, an object is accepted only after K consecutive verifier passes. With $K = 1$, many accepted objects fail a new verifier check.



Answer scoring via LLM judges

- An LLM judge scores answers against references along 7 axes, then produces the final score. We use the average of two judges to reduce judge-specific biases.
- We show the judges are well calibrated and reliably detect rubric violations.
- The judges rank models similarly and closely agree in absolute scores.
- A small human study (with simplified rubric, hence higher scores) shows the judges and human evaluator rank models similarly.

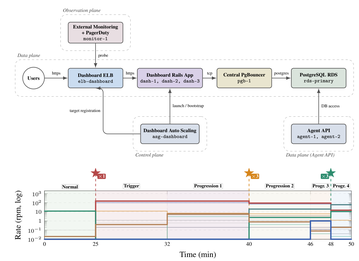


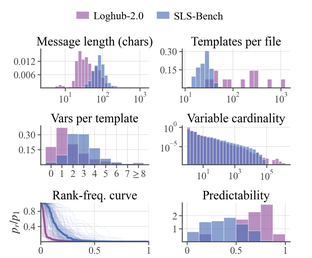


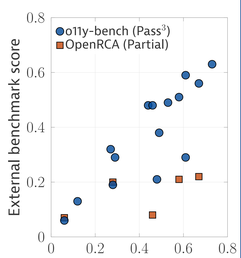
Model	Judge	Human
Nemotron 3 Nano	0.00	0.10
DeepSeek V3.2	0.10	0.20
GPT-5.4 Nano	0.38	0.35
Opus 4.6	0.53	0.65
GPT-5.4	0.63	0.95

Realism of the synthetic telemetry

- High-level realism (system structure, failure type, dynamics, and observables) is promoted by report-grounded incident models. On average: 6 system components and 8 edges, 16 hosts, 27 log templates, 4 failure phases, 48 minutes of observations.
- Log-level realism is evaluated by comparison to 14 production log streams from Loghub-2.0 dataset, showing a close match on many relevant metrics.
- Comparisons to third-party benchmarks show SLS-Bench scores are predictive of performance on SRE tasks beyond log summarization.

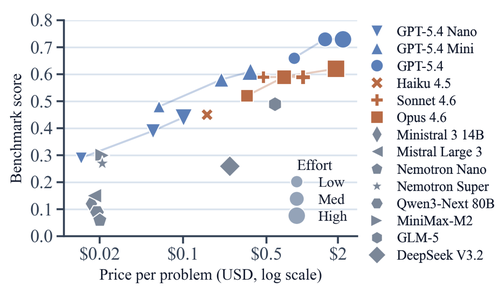


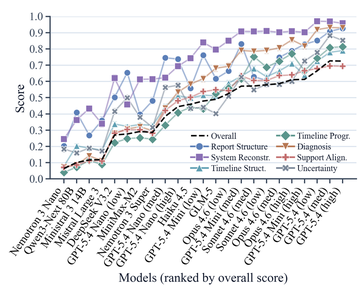




Main results

- We evaluate 14 LLMs at different reasoning efforts. The best, GPT-5.4 (high), scores 0.73, close to a "good match" (0.75), at $2.2 and 10 minutes per task on average.
- Scores tend to grow with cost, and within each model size lower reasoning effort is the most cost-efficient. Open-weight models lag behind; the best, GLM-5, reaches the score of 0.49.
- Best models identify the core fault chain well ("diagnosis" near 1.0), but "timeline progression" (recovering failure phases) and "support alignment" (finding and interpreting log evidence) plateau at 0.7–0.8.





Code: github.com/dynatrace-research/sls-bench
Dataset: huggingface.co/datasets/dynatrace-oss/SLS-Bench