SRE-Zero: Environment-Grounded Evaluation for Reliable Tool-Using Agents

Devaansh Pathak

Draft v1.5 – May 23, 2026

Abstract

SRE-Zero is an early-stage simulation benchmark for studying reliable tool-using agents in site reliability engineering incident-response workflows. Agents must inspect simulated service state, gather evidence, apply minimal remediations, and submit a final incident resolution under a step budget. This v1.5 draft updates the first technical report with a broader 25-task preliminary baseline sweep across random, scripted, prompting-only, ReAct-style, open-source, and frontier-model baselines. The current results remain preliminary and are intended to validate the benchmark design, not to provide final model rankings. The main finding is that evidence gathering and final resolution remain separable: several agents gather relevant evidence but do not reliably convert that evidence into the correct fix and final resolution.

1 Status

This is an early public draft. The current results are preliminary and are intended to validate the benchmark design, not to provide final model rankings.

The v1.5 update uses one seed, one episode per LLM task, a small model set, and provider-backed API calls that produced errors for some runs. These results should therefore be interpreted as a benchmark validation sweep rather than a stable leaderboard.

2 Introduction

Tool-using agents are increasingly evaluated on workflows that require external state inspection and multi-step action. In operational settings, reliability is not only final answer correctness. A useful incident-response agent should choose appropriate tools, gather relevant evidence, avoid invalid actions, apply minimal remediations, and resolve the incident within a finite step budget.

SRE-Zero studies this problem in a controlled simulated environment. The environment does not execute shell commands, control real infrastructure, or perform live remediation. It exposes structured actions over simulated service state and returns typed observations, rewards, terminal state, and metrics.

Contributions in this draft.

1. A simulation-only incident-response environment with typed actions and observations.
2. A 25-task suite spanning web server, database, cache, message queue, and load balancer incidents.

3. A partial-credit reward and metrics that separate success, evidence coverage, remediation quality, invalid actions, and efficiency.
4. A preliminary baseline sweep across deterministic, prompting, ReAct, open-source, and frontier-model agents.

3 Environment

Each episode begins with an alert and a hidden incident configuration. The agent can inspect simulated logs, metrics, status, and configuration for five services:

- web server
- database
- cache
- message queue
- load balancer

The action space is structured:

- `inspect_logs(service)`
- `inspect_metrics(service)`
- `check_status(service)`
- `inspect_config(service, key?)`
- `restart_service(service)`
- `update_config(service, key, value)`
- `resolve_incident(root_cause, fix)`
- `escalate(reason)`

Invalid actions return controlled error observations and are counted in the metrics. They do not crash the environment.

4 Task Suite

The v1.5 sweep uses the expanded 25-task suite. The suite contains easy, medium, and hard incidents. Tasks include service crashes, configuration regressions, resource saturation, misleading symptoms, distractor logs, and noisy metrics. Examples include cache crashes, database connection pool exhaustion, low web timeouts, queue backlogs, load balancer health-check misconfiguration, and incidents where the visible web error is caused by a database or cache root cause.

5 Reward and Metrics

SRE-Zero uses partial-credit rewards so that incomplete but useful behavior is measurable. The reward and marks accounting separate:

- relevant evidence gathered
- correct root cause
- correct remediation
- correct final resolution
- efficiency
- invalid actions
- wrong remediation
- premature resolution
- distractor failure

This design is meant to reveal more than success rate. An agent that gathers evidence but fails to resolve is different from an agent that cannot produce valid actions.

6 Baselines

The v1.5 sweep evaluates:

- a random baseline
- a scripted expert baseline
- prompting-only `openai/gpt-5-mini`
- ReAct-style `openai/gpt-5-mini`
- ReAct-style `anthropic/claude-sonnet-4.6`
- six open-source or open-weight model configurations
- three frontier model configurations

Deterministic baselines use five episodes per task. LLM baselines use one episode per task. All results in this draft use seed 0.

7 Results

Table 1 reports the preliminary 25-task sweep.

Baseline	Model	Marks	Success	Reward	Evidence	Errors
scripted	scripted	93.4	1.00	0.943	1.00	0
frontier	gpt-5.5	57.4	0.52	0.527	0.86	2
frontier	Claude Opus 4.7	48.3	0.40	0.470	0.68	5
ReAct	Claude Sonnet 4.6	46.1	0.36	0.417	0.81	0
open-source	Mistral Small 3.2 24B	24.9	0.04	0.099	0.79	0
ReAct	gpt-5-mini	18.1	0.00	0.012	0.66	24
open-source	Nemotron 120B free	17.3	0.00	0.039	0.61	5
prompting	gpt-5-mini	16.2	0.00	0.014	0.55	20
open-source	gpt-oss-20b free	11.3	0.00	0.003	0.31	20
random	random	5.4	0.00	0.004	0.04	0
frontier	Claude Sonnet 4.6	5.3	0.00	0.000	0.01	25
open-source	Llama 3.3 70B free	5.0	0.00	0.000	0.00	25
open-source	Qwen3 Next 80B free	5.0	0.00	0.000	0.00	25
open-source	Gemma 4 26B free	5.0	0.00	0.000	0.00	25

Table 1: Preliminary SRE-Zero 25-task baseline sweep. Results use seed 0. LLM runs use one episode per task. Error counts include agent/provider-level failures and should be interpreted cautiously.

8 Analysis

The scripted expert solves the benchmark and receives high evidence, reward, success, and validity credit. The random baseline remains close to the floor, with 5.4 marks and 0.04 evidence coverage.

The strongest frontier run is `openai/gpt-5.5`, which achieves 57.4 marks, 0.52 success, and 0.86 evidence coverage. This suggests the agent often finds relevant evidence but still misses a substantial fraction of final resolutions.

The ReAct-style `anthropic/claude-sonnet-4.6` run reaches 0.81 evidence coverage and 0.36 success with no agent-level errors. This is a useful intermediate behavior: the agent is far from the scripted upper bound but clearly above the random floor.

The clearest repeated signal is that evidence gathering and final resolution are separable. For example, `mistralai/mistral-small-3.2-24b-instruct` reaches 0.79 evidence coverage but only 0.04 success. Prompting and ReAct `gpt-5-mini` runs also collect evidence but fail to resolve any tasks in this sweep.

9 Limitations

This draft has several important limitations:

- one seed
- one episode per LLM task
- a limited model and provider set
- provider errors in several API-backed runs
- simulated services rather than real infrastructure
- no human SRE baseline

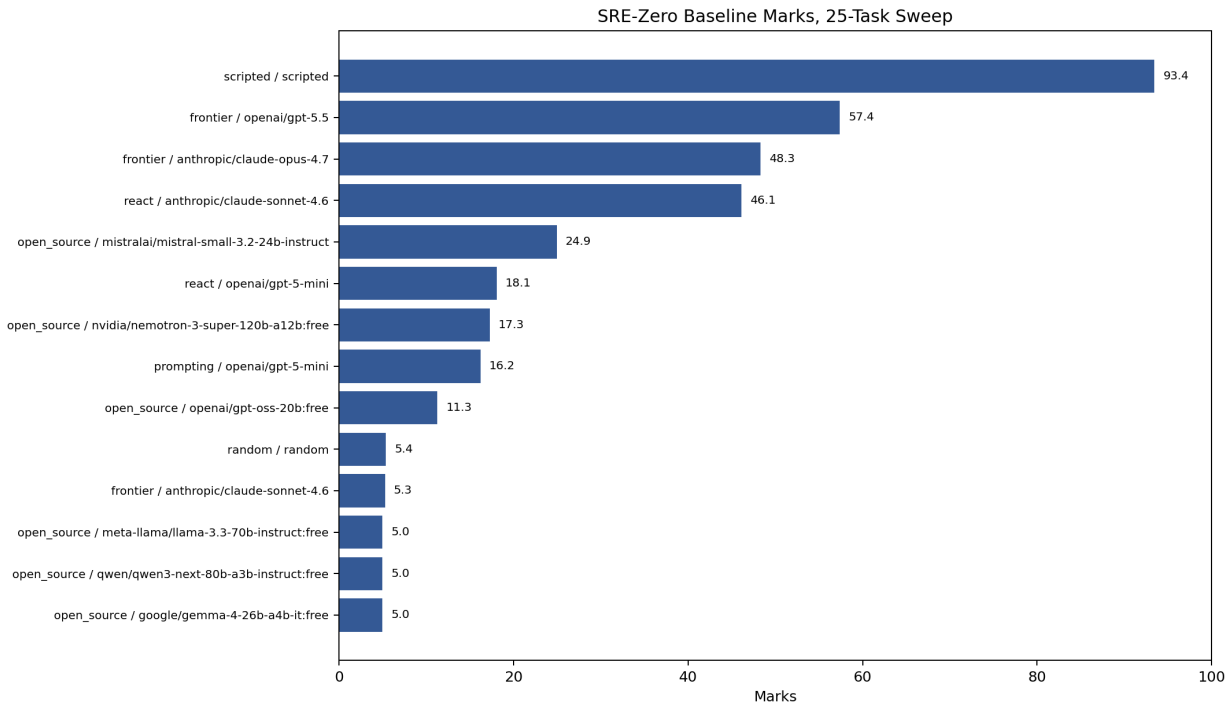

Figure 1: Overall marks for the 25-task preliminary baseline sweep.

- no confidence intervals

The current results should not be used as final model rankings. The primary claim is that SRE-Zero produces inspectable differences between agent strategies and can separate evidence gathering, action validity, remediation quality, and final resolution.

10 Conclusion

SRE-Zero is a small but growing benchmark for environment-grounded evaluation of reliable tool-using agents. The v1.5 preliminary sweep expands the task suite and shows a useful pattern: agents can gather evidence without reliably resolving incidents. This supports the benchmark direction and motivates more careful future evaluations with more seeds, stronger provider controls, and larger task suites.

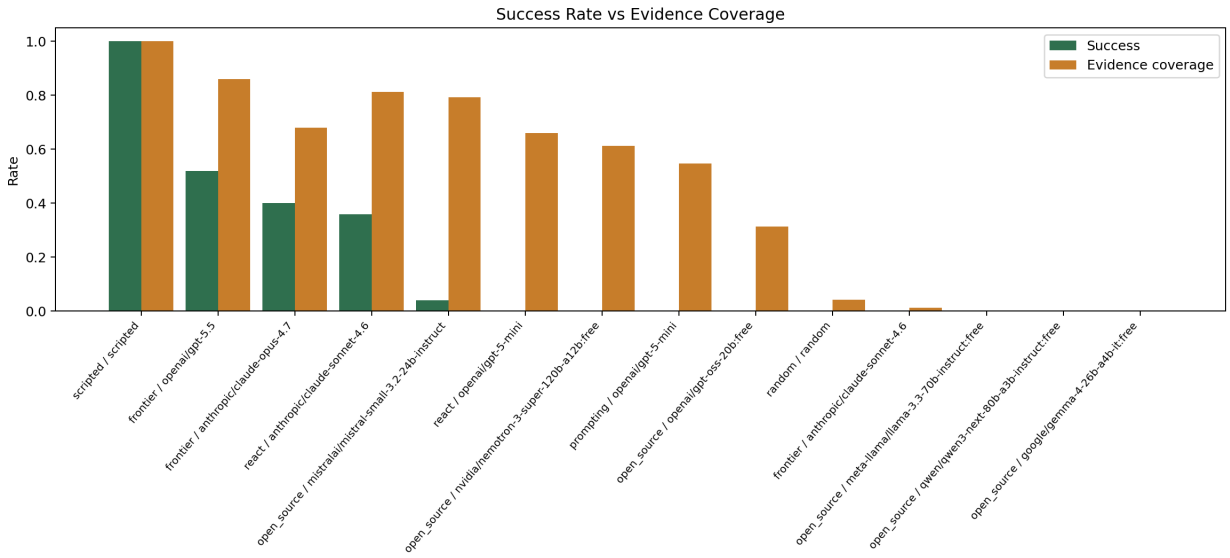

Figure 2: Success rate and evidence coverage are not the same capability.

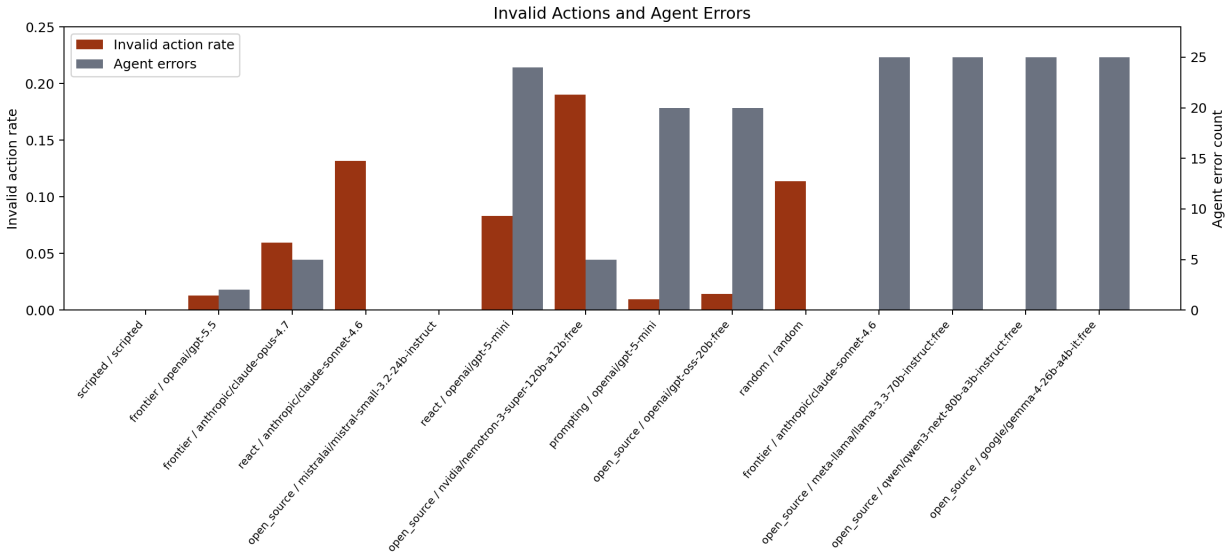

Figure 3: Invalid actions and agent errors are tracked separately. Some runs failed primarily due to provider or output errors rather than environment difficulty alone.

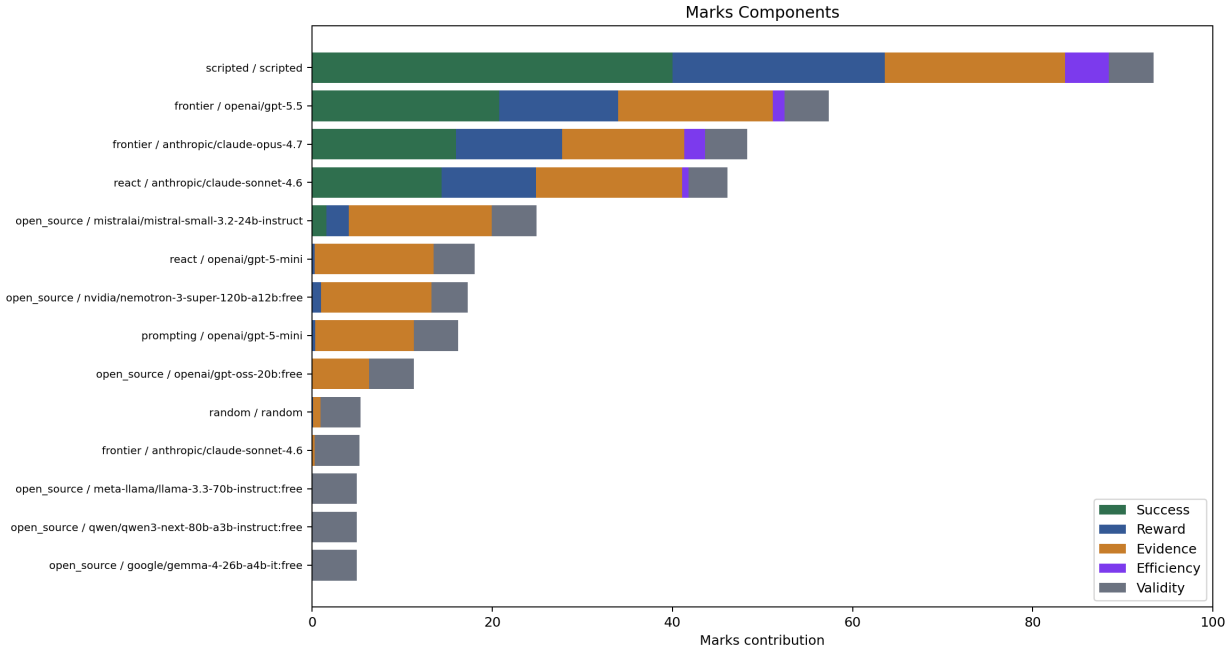

Figure 4: Marks components separate success, reward, evidence, efficiency, and validity.

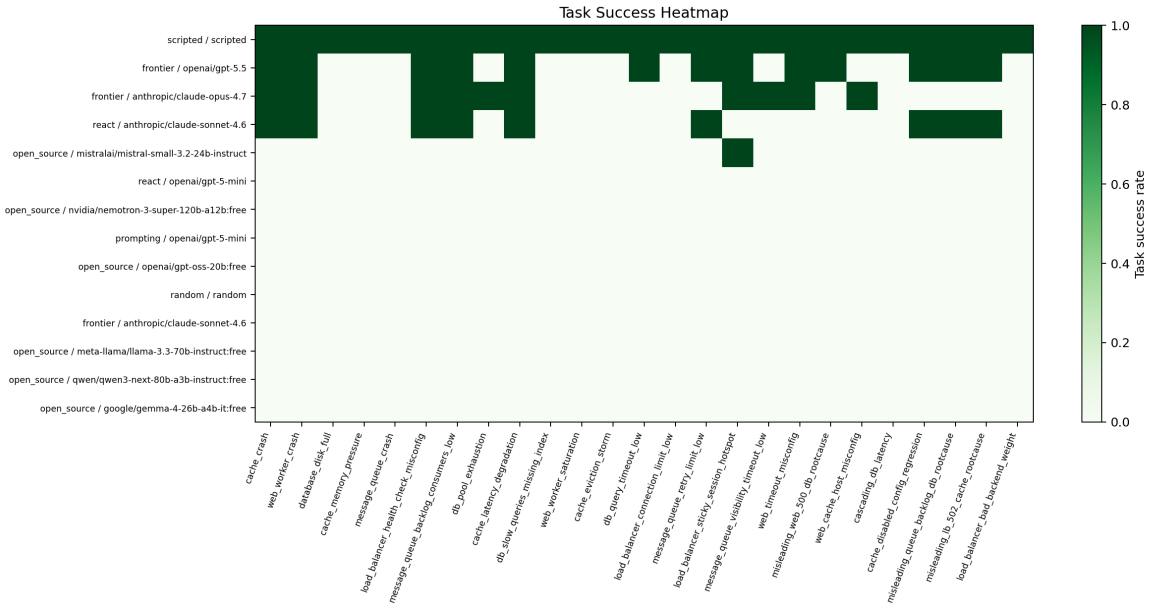

Figure 5: Task-level success rates show that different baselines fail on different subsets of the suite.