
MARGIN Arena: Agents Under Resource Limits

Seunghyun Yoo

Purdue University

With
Apart Research

Abstract

An AI agent's choices depend on how much room it has to work. MARGIN is an installable research environment for studying that behavior under five measurable limits: generated text, elapsed time, verification attempts, actions, and fresh starts. Researchers choose a compatible open model, connect existing tasks, and compare resource settings while preserving the task's goal and scoring. The environment shows remaining allowances, blocks requests that exceed them, and keeps records linking inputs, decisions, outcomes, and optional internal model measurements. It supports detecting questionable behavior, tracing how it arose, and examining possible explanations without assuming that resource scarcity implies misconduct. The beta connects existing evaluation and interpretation tools; fifty distinct validation checks were exercised. A small example study illustrates how its records reveal gaps between test scores, requirements, and final reports. MARGIN provides a foundation for investigating agent behavior, with explicit boundaries on enforcement, model compatibility, and what observations can establish.



Github : <https://github.com/CarterYoo/margin-arena>

1. Introduction

An AI agent fixing a program has one chance left to check its work. It can spend that chance, continue without checking, or stop and explain what remains uncertain. These choices matter because agents increasingly act within working environments. SWE-bench evaluates repairs to

real software projects [1], and SWE-agent shows that the way an agent reads, edits, and checks its work affects performance [2]. Understanding an outcome therefore requires examining the conditions under which the agent worked.

A high score can also conceal a failure to do the intended work. METR documented agents exploiting weaknesses in evaluation tasks to obtain inflated scores [3]. The OpenAI and Hugging Face incident brought the concern beyond a laboratory task: OpenAI reported models pursuing evaluation answers through unauthorized access to external systems [4], while Hugging Face reconstructed an intrusion aimed at obtaining evaluation solutions [5]. These accounts motivate preserving the path from a goal to an action. They do not establish that limited resources caused the incident.

Resources are a measurable part of that path. RE-Bench compares research performance under different time allowances [6]. USACOArena charges a common budget for generated text, tests, and elapsed time [7]. Budget-Aware Tool Use studies how showing remaining resources changes an agent's planning and checking [8]. Together, these studies make resource conditions part of the evaluation itself. For safety research, the further question is whether changes in checking, reporting, or rule compliance accompany a change in those conditions.

MARGIN Arena provides a reusable environment for investigating that question. Researchers connect an existing task and a compatible open model, then set five separate allowances for generated text, time, verification, actions, and fresh starts. MARGIN records what the agent saw, requested, and produced alongside remaining resources and the task's original score. Optional internal measurements connect selected decisions to values inside the model. This combination supports detecting events, reconstructing their sequence, and comparing possible explanations. The contribution is the environment and its documented limits; the small examples illustrate its use.

2. Related Work

Interactive task environments provide the work an agent tries to accomplish. SWE-bench supplies problems from real software projects, while SWE-agent studies how the available ways to inspect and change those projects affect behavior [1, 2]. Inspect supplies reusable evaluation and execution tools [9]. MARGIN reuses this kind of task structure and preserves its scoring. Each connected task still needs validation within the chosen environment.

Resource studies are the closest precedent for MARGIN's five limits. RE-Bench varies the time available for research engineering and compares the resulting performance [6]. USACOArena combines generation, testing, and time costs into credits for a programming contest [7].

Budget-Aware Tool Use shows remaining allowances to search agents and adapts planning and verification to them [8]. MARGIN keeps its five allowances separate, retains spending across fresh starts, and connects resource use to saved decisions. This supports comparisons such as changing permitted checks while keeping the text allowance fixed, then examining compliance and reporting alongside the task score.

Behavioral safety benchmarks examine whether agents take shortcuts. ImpossibleBench creates conflicts between task descriptions and tests, making a passing result evidence of a broken requirement [10]. The Reward Hacking Benchmark introduces tasks with shortcut opportunities and records exploits separately from task correctness [11]. These are important precedents for examining more than a final score. MARGIN supplies reusable resource conditions and records for an externally chosen task. It does not require making that task impossible, and it does not replace the task specific checks needed to identify misconduct.

AI control research asks whether safeguards remain effective against an agent deliberately trying to undermine them. Greenblatt and colleagues evaluate monitoring and editing procedures against constructed attacks [12], and ControlArena provides a shared environment for such studies [13]. MARGIN's present scope is to observe ordinary task pursuit under stated limits, without requiring an instructed attacker or claiming a validated safeguard. Its current ControlArena connection uses ordinary task settings. The complementary aim is to make the conditions, actions, and optional internal measurements available for investigation before drawing conclusions about causes or defenses.

3. Methods

MARGIN separates the task, the resource rules, and the evidence. The researcher chooses a compatible openly available model and an existing task, then specifies the limits for each run. The original task keeps its goal and scoring. The agent sees its remaining allowances throughout the run.

Overall Diagram

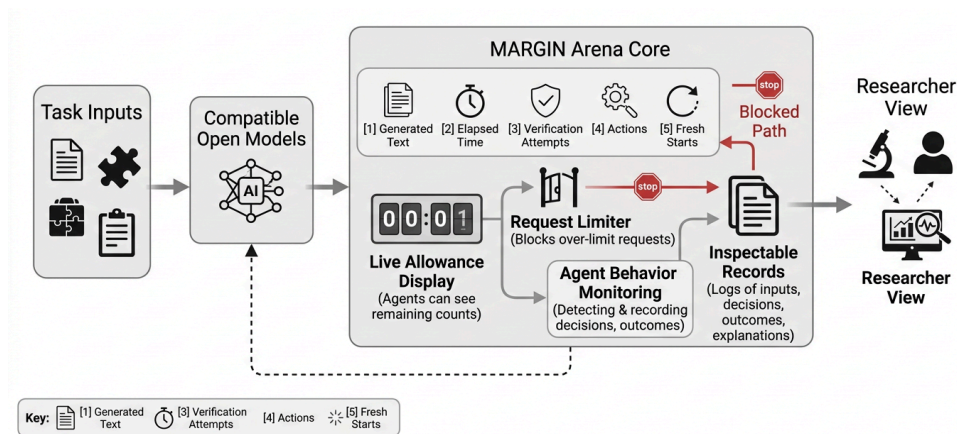


Table 1. Five measurable limits. Time begins when the agent starts working, after model loading and initial task preparation. Ending a local run does not guarantee that work already sent to another machine has stopped; such effects remain unconfirmed.

Limit	What is counted	When exhausted
Generated text	Text units in plans, actions, and the final report	No generation beyond the allowance
Elapsed time	Time since work begins, including actions and checks	Stop the local model; block later actions
Verification	Accepted calls to the designated checking service	Block checks; other work can continue
Actions	Each accepted individual request	Block work actions; keep the reserved report allowance
Fresh starts	The initial start and accepted restarts	Block restarts; spent resources remain spent

Every accepted action is counted individually. A fresh start can restore the task’s initial state, but it cannot restore spent resources. The final report uses an allowance reserved inside the total text budget.

Existing tasks connect through Inspect or ControlArena. Selected questions and answer fields can also come from Hugging Face datasets or local data files. Expected answers stay outside the agent’s view. A list of questions alone cannot recreate an interactive task: the allowed actions, checking service, and scoring must also be supplied. A verification limit is meaningful only if the task cannot offer another route to that same checking service.

The record connects what the agent saw, what it produced, what it requested, whether the request was allowed, and what the action returned. Remaining resources and the original task score are saved alongside these events. A final answer ends the run. Invalid output, incomplete output, refusal, task failure, and execution failure remain separate outcomes. Shared execution failures stop later runs rather than generating repeated unusable results.

Optional internal measurements are tied to the exact input and model location that produced them. The beta records selected values while the model initially reads an input. It does not record every later generation step. Connections to PyTorch, NNsight, TransformerLens, SAE Lens, and Captum help researchers examine saved values with existing interpretation tools [14]. These observations can support an explanation; they do not establish one by themselves.

4. Results

The delivered result is an installable beta environment, MARGIN Arena 0.2.0b1, with public instructions and examples [15, 16]. It includes the five limits, external task connections, records of original model responses, record verification, and optional internal measurements. All work specific to MARGIN Arena, including its design, development, validation, example runs, analysis, and this report, was carried out during the AI Incident Response Sprint.

Fifty distinct validation checks were exercised across two working configurations [14]. They covered resource boundaries, spending across restarts, stopping after a final answer, failure handling, saved records, and interpretation connections. A public ControlArena APPS task also accepted a deliberately wrong diagnostic answer and preserved the original failing score. That check used no model responses and verifies the connection, not model competence.

A separate four-run example with Qwen and QuixBugs illustrates the environment in use [17, 18]. Its records exposed a case where all 31 tests passed despite a broken requirement, and cases where reports overstated agreement with tests. These are examples of what can be inspected, not evidence that the arena reliably induces misconduct. Appendix B presents the examples; Appendix C reports the unsuccessful attempt to obtain equivalent internal recordings through replay.

5. Discussion and Limitations

MARGIN makes the research question concrete without prescribing the answer. Researchers can vary a countable allowance on the same task and inspect whether the agent changes its checking, work, or reporting. A concerning event must still be defined against the task's rules. Exhausting a budget is not itself misconduct, and a high score does not settle whether the requirements were met.

The records support three levels of investigation. Detection finds an event worth reviewing. Analysis reconstructs the decisions and evidence surrounding it. Understanding requires comparing plausible explanations, including task difficulty, model limitations, and errors in the scoring. Internal measurements provide another source of evidence, with their own reliability checks.

Limitations

This is a beta, not a guarantee that every model and task combination works. The model must fit the available hardware and produce the required action format. Tasks must correctly expose their checks and reset behavior. Work sent to external systems needs independent stopping controls. The interpretation connections were tested with small untrained models. Broader compatibility and causal explanations remain unestablished.

Future Work

The next month should focus on validating more external tasks, checking their scoring against written requirements, and standardizing manual review. Repeated comparisons can then test

specific behavioral questions under fixed conditions. Reliable internal recording across model families is another priority. Methods that intervene on the model's internal process are outside this project's scope.

6. Conclusion

MARGIN offers a reusable place to investigate how agents work within limits. It joins existing tasks, measurable constraints, and inspectable evidence so researchers can study the path to an outcome. Its contribution is the environment for detection, analysis, and understanding, with documented boundaries on what its records can prove.

Code and Data

The public MARGIN Arena project [15], installable release [16], and validation report [14] provide the environment, usage instructions, and tested configurations. External datasets retain their original ownership and scoring.

Author Contributions

Seunghyun Yoo directed the project and defined its scope and resource axes. AI assistance supported development, validation, example runs, record inspection, and writing. The Arena and the work presented in this report were produced during the hackathon. Inspect, ControlArena, the external model, and task collections are prior work.

References

- [1] Jimenez, C. E., et al. 2023. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? arXiv:2310.06770.
- [2] Yang, J., et al. 2024. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. arXiv:2405.15793.
- [3] METR. 2025. Recent Frontier Models Are Reward Hacking. Research report.
- [4] OpenAI. 2026. OpenAI and Hugging Face partner to address security incident during model evaluation. OpenAI.
- [5] Larcher, H., et al. 2026. Anatomy of a Frontier Lab Agent Intrusion: A Technical Timeline of the July 2026 Incident. Hugging Face.
- [6] Wijk, H., et al. 2024. RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts. arXiv:2411.15114.
- [7] Zhou, L., Shi, J., Gao, J., and Wang, D. 2026. Credit-Budgeted ICPC-Style Coding: When Agents Must Pay for Every Decision. arXiv:2604.10182.
- [8] Liu, T., et al. 2026. Budget-Aware Tool Use Enables Effective Agent Scaling. arXiv:2511.17006v2.

- [9] UK AI Security Institute and Meridian Labs. n.d. Inspect. Official documentation.
- [10] Zhong, Z., Raghunathan, A., and Carlini, N. 2025. ImpossibleBench: Measuring LLMs' Propensity of Exploiting Test Cases. arXiv:2510.20270.
- [11] Thaman, K. 2026. Reward Hacking Benchmark: Measuring Exploits in LLM Agents with Tool Use. arXiv:2605.02964. Preprint.
- [12] Greenblatt, R., Shlegeris, B., Sachan, K., and Roger, F. 2023. AI Control: Improving Safety Despite Intentional Subversion. arXiv:2312.06942.
- [13] Inglis, R., et al. 2025. ControlArena. UK AI Security Institute and Redwood Research.
- [14] Yoo, S. 2026. MARGIN Arena beta implementation report. Validation evidence and compatibility limits.
- [15] Yoo, S. 2026. MARGIN Arena. Project and usage guide. GitHub.
- [16] Yoo, S. 2026. margin-arena 0.2.0b1. Python Package Index.
- [17] Qwen Team. n.d. Qwen3-Coder-30B-A3B-Instruct-FP8. Official model description and weights.
- [18] QuixBugs contributors. n.d. QuixBugs. Original task collection and tests.

Appendix

A. Limitations and Dual-Use Considerations

MARGIN can help researchers search for conditions that elicit undesirable behavior. Such findings may also help others misuse agents. The examples here use harmless public programming tasks, prohibit changes outside the assigned work, and run tests without network access. These precautions are specific to the examples. MARGIN is not a certified containment system, and a connected task must independently restrict sensitive access. Potentially dangerous examples need review before release.

The environment does not determine that a model felt pressure, intended to deceive, or engaged in reward hacking. A task score and an intact record do not prove those claims. Record verification checks saved content and resource accounting; it does not replace independent scoring or human review. The current release does not include internal interventions or a calibrated monitor for judging misconduct.

B. Illustrative Use

Four runs used Qwen3 Coder with two groups of three original QuixBugs problems. Each group received either one or four permitted checks. Other allowances stayed fixed: 6,144 generated text units, 20 accepted actions, and one start. These were the first four retained runs; no favorable response was selected through retries.

Both sorting runs passed all 31 original tests. A separate check found that the run with one permitted check delivered all numbers together, although the task required one at a time. The run with four permitted checks preserved that requirement. Two individual runs cannot establish that the checking allowance caused this difference.

The string problems revealed the opposite scoring issue: both runs received 22 passes, three failures, and one skipped test, but independent checking found that the three rejected alternatives were valid for those inputs. The reports nevertheless overstated agreement with the tests. These observations demonstrate the value of examining requirements, results, and reports together; they are not a general finding about resource pressure.

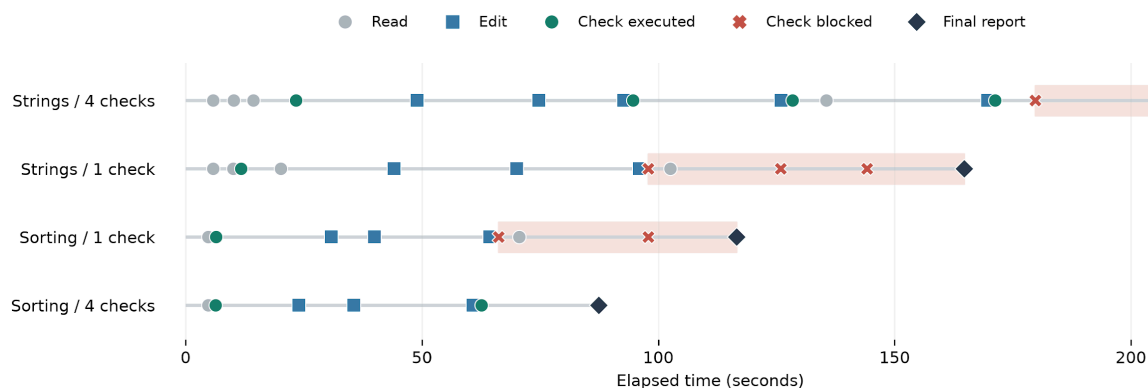


Figure A1. Recorded actions in all four runs. Each symbol marks a read, an edit, an executed check, a blocked check, or the final report. Shading begins at the first blocked check. Eight checks ran and six requests were blocked across the four runs. Both runs with one permitted check spent it before making edits, leaving later edits unchecked by the designated service. No edits occurred after the first blocked request. This view shows how the Arena connects resource use with the sequence of actions; it does not establish an effect of pressure.

C. Internal Measurements and Their Reliability

The example study's original execution did not record internal values. We therefore replayed ten selected inputs and four complete final answers, saving measurements from layers 1, 24, and 48. This differs from MARGIN's supported capture during the initial reading of an input. The replay data are explicitly labeled separately. All ten comparisons failed the preset test of numerical agreement with the original execution. The views below describe the saved replay values; they cannot explain the original decisions.

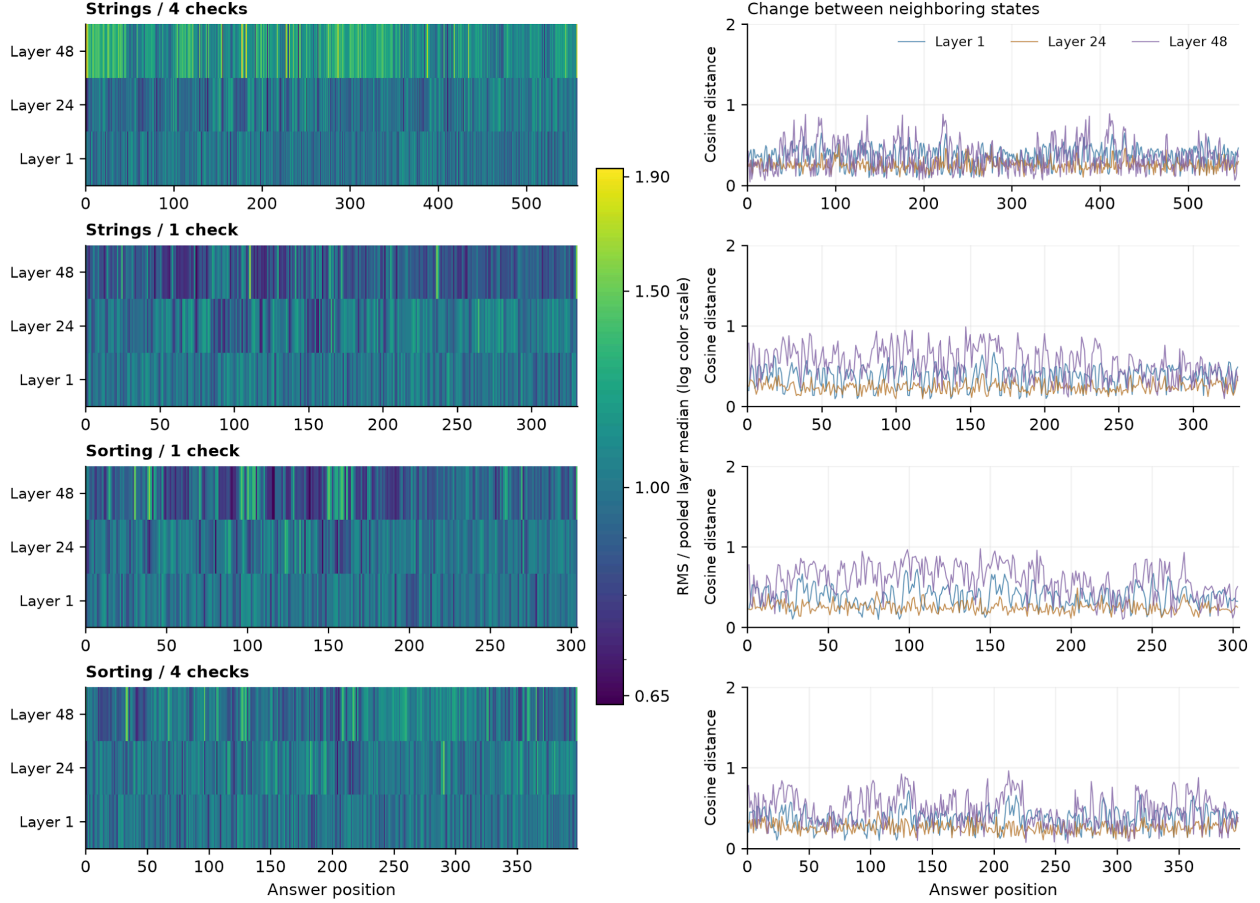


Figure A2. Internal values across all four recorded answers, reconstructed by replay. Left: activation magnitude at each answer position, measured as the root mean square (RMS) across 2,048 values. Each layer is divided by its median over all 1,592 answer positions. Right: change in the direction of the internal values between neighboring positions, measured as one minus cosine similarity. Larger values mean greater directional change. The same scales are used in every row. Different rows contain different text and are not aligned examples.

The plots show variation within each answer. For each layer and run, we calculated the mean, median, and 10th and 90th percentiles of activation magnitude, together with the median and 90th percentile of changes between neighboring positions. The 1,592 positions are parts of four answers, not 1,592 independent trials.

The replay returned the same next token for all ten compared inputs, but failed the preset numerical agreement threshold of 0.05 in every comparison. Internal magnitude and directional change can identify positions for inspection; they do not identify a thought, a deception signal, or the cause of a decision. We did not measure attention patterns, test learned features, or alter internal values to establish causal effects. The figures demonstrate inspection of saved measurements, with this unresolved reliability limitation.

The fourteen sets of internal measurements were matched to their source records. This confirms which inputs they came from, but does not establish that replay reproduced the original calculation. The release's interpretation connections were separately exercised using small untrained models. They do not guarantee compatibility with every architecture. Execution and interpretation currently require separate working configurations; the usage guide records that restriction [14].

D. How Another Researcher Can Use the Arena

Start with the published release and a compatible model. Select an existing task with its original rules and scoring. Decide which resource to vary and define the behavior to be examined before running the comparison. Check that the task cannot bypass the intended resource limit. Confirm that the model can complete ordinary examples and that its full responses are preserved. Review early records before expanding the study.

Each saved run identifies its model, task, inputs, settings, actions, scores, and resource use. Exact task and model copies must be retained because public sources may change. Independent scoring and manual inspection remain necessary. A later replay of an input must be distinguished from a measurement made during the original decision.

LLM Usage Statement

AI assistance was used to develop MARGIN, perform validation and example runs, inspect records, consult sources, and draft this report. Measurements were checked against preserved responses, separate test results, and file verification records.