

Peer Review

Review of: "MedAgentBench: A Realistic Virtual EHR Environment to Benchmark Medical LLM Agents"

Evgenii Arsentev¹

1. Independent researcher

Summary

The authors build MedAgentBench, a benchmark for LLM agents acting inside an electronic health record. The pieces are: 300 tasks written by two internal medicine physicians who are also authors (Section 2.1), 100 deidentified Stanford patient profiles carrying 785,207 records in total (Table 2, row "Number of records (total)"), a HAPI FHIR JPA server packaged as a Docker image (Section 2.3), and a small orchestrator that exposes nine FHIR functions and caps each episode at 8 rounds (Section 2.4.3). Twelve models are scored under pass@1 with task success rate as the only metric (Section 2.4.1). The headline result is Claude 3.5 Sonnet v2 at 69.67% overall (Table 3, row "Claude 3.5 Sonnet v2", column "Overall SR"), which the authors read as an unsaturated benchmark with room to grow. The artifact is the real contribution here, and it is a good one: a FHIR-shaped environment plus clinician-written tasks is exactly what this field has been missing. My concerns are almost entirely about the measurement layer stacked on top of it.

Two things I checked first, because reviewers often flag them wrongly. Table 2 adds up: 563,426 Observation + 124,969 Procedure + 74,821 Condition + 21,991 MedicationRequest = 785,207, matching the stated total exactly. And Table 3 is internally coherent: with 150 query and 150 action tasks (Section 2.5.1), every row's Overall SR is the mean of its Query SR and Action SR to rounding, and every row of Table 4 reproduces its Overall SR under group sizes of 90 easy, 120 medium, 90 hard. The bookkeeping is careful. The inferential claims built on it are not.

Does the claim hold on the data

Partly. "Current models show non-trivial but unreliable agent performance" survives contact with the data. The leaderboard ordering does not, and neither does one claim in the Abstract.

First, the ranking. Every number in Table 3 is a single pass@1 run at temperature zero (Section 2.4.2) over 300 tasks. No repeats, no seeds, no intervals, no test. By my own arithmetic — the paper reports none — a 95% binomial interval around 69.67% at $n=300$ is roughly ± 5.2 points, and around GPT-4o's 64.00% (Table 3, row "GPT-4o", column "Overall SR") roughly ± 5.4 . Those overlap heavily, and so do DeepSeek-V3 at 62.67% and Gemini-1.5 Pro at 62.00% (same column). An unpaired two-proportion test on Claude versus GPT-4o gives $z \approx 1.48$, $p \approx 0.14$. The comparison is actually paired — same 300 tasks for every model — so McNemar on per-task outcomes is the right instrument and would be more powerful, and the authors already have the per-task data to run it. As it stands, the sentence "The best model (Claude 3.5 Sonnet v2) achieves a success rate of 69.67%" in the Abstract asserts a ranking the design cannot resolve among the top four.

Second, the headline number hides the half of the benchmark that matters most for safety. Claude 3.5 Sonnet v2 leads on Overall SR and on Query SR (85.33%, Table 3), but on the Action SR column it comes fifth: Gemini-1.5 Pro 71.33%, Qwen2.5 64.00%, GPT-4o 56.00%, DeepSeek-V3 54.67%, then Claude 3.5 Sonnet v2 at 54.00%. In a benchmark whose stated motivation is that "even a single incorrect action or response can have significant consequences" (Section 2.4.1), presenting one averaged number as "the best model" inverts the paper's own safety argument. The Abstract and Section 2.5 should lead with the split.

Third, and this is the sharpest problem: on action tasks, the environment is never actually written to. Section 2.4 states that only GET requests are sent to the server because startup takes about 90 seconds. Section 2.4.3 confirms that a POST gets "a simple sanity check to make sure the payload data is JSON-loadable," after which the harness indicates success to the agent regardless. So 150 of the 300 tasks are graded on a payload string against hand-written rules (Section 2.4.1), never against the resulting server state, and the agent is always told its write succeeded. Three consequences. The FHIR server is not exercised as an environment for the action half — it is a read-only fixture, which weakens the Abstract's claim that the setup "can be easily migrated into live EMR systems," since the write path is never validated against the server that would do the migrating. The agent never faces a rejected write, which is the single most important recovery behavior in a real EHR. And FHIR admits many semantically

equivalent payloads for the same clinical intent, so the rules' recall is a free parameter that no reported number constrains. A 90-second startup is a solvable engineering problem — DB snapshot and restore, or a transaction rolled back per task — and solving it would materially raise the value of the benchmark.

Fourth, a large share of what Table 3 measures is output-format compliance, not clinical agent competence. Section 2.5.2 reports Gemini 2.0 Flash emitting invalid actions in 54% of cases, largely because it wraps output in a `tool_code` block. Section 2.4.3 says the harness strips those separators — for Gemini. That is a model-family-specific repair applied to one vendor and not, as far as the text says, to anyone else, which makes the comparison uneven by construction. Meanwhile, the orchestrator asks every model to emit bespoke plain text ("`GET url?param_name1=...`" / "`POST url [json]`" / "`finish([...])`", Appendix A.2) although all twelve providers expose native structured tool calling. Gemini 2.0 Pro landing at 18.00% overall with 10.67% Action SR (Table 3), far below Qwen2.5 at 51.33%, reads much more like a parser mismatch than a capability gap, and the paper does not investigate it. Without a format-compliance ablation, readers cannot tell how much of the leaderboard is medicine and how much is string handling.

Fifth, an Abstract claim with no supporting table. "There is significant variation in performance across task categories" appears in the Abstract, again in Section 1, and again at the end of Section 2.5. But no per-category results exist anywhere in this version. Table 3 splits query versus action; Table 4 splits by difficulty; Table 1 lists seven broad categories, and the text says there are ten specific ones. There is no table of success rate by category. The word "significant" is also doing statistical work it has not earned, which matters more given the per-category *n* would be about 30.

Sixth, the difficulty construct in Appendix A.3 is defined purely by step count and does not behave monotonically. Gemini-1.5 Pro scores Easy 82.22%, Medium 45.83%, Hard 63.33% (Table 4, row "Gemini-1.5 Pro"); Llama 3.3 scores Medium 38.33% below Hard 46.67% (same table). If "hard" is reliably easier than "medium" for two of twelve models, step count is not tracking difficulty, or the medium bucket contains a systematic format trap. Either way, it needs a sentence, ideally a per-category error breakdown.

On sampling: 300 tasks sound like a lot, but Table 1 shows templated instructions with {MRN}, {name}, {DOB} placeholders, and 300 tasks across 10 categories is 30 per category. Items sharing a template and a grader are correlated, so the effective sample size is well below 300, and my interval estimate above is, if anything, optimistic. The paper should report how many distinct templates underlie the 300 items and cluster its intervals accordingly.

Reproducibility

The environment and code are public (Section 2.3, Abstract), which is more than most benchmark papers offer, and the Docker image gives a real chance at replication. But the model-side reporting will not support re-running these exact numbers.

No model snapshot is named. Section 2.4.2 lists "GPT-4o", "Gemini 2.0 Pro", "o3-mini", "Claude 3.5 Sonnet v2" and so on without dated versions, so "GPT-4o" could be several different models, and any rerun six months out will silently measure something else. No evaluation dates are given. For o3-mini, the reasoning effort setting is absent, which matters a great deal for a reasoning model, and Section 2.4.2 notes temperature zero was applied to every model "except o3-mini" — so that single 51.67% (Table 3) is one draw from a stochastic model compared head to head with greedy decodes. The five open-weights models are served through Together AI's serverless API (Section 2.4.2) with no precision or quantization stated; a served endpoint's weights are not necessarily the released weights, which turns the "gap between closed and open-weights LLMs" claim in Section 2.5 into a claim about specific hosting endpoints. No seeds anywhere, no retry policy for API errors, no statement of what happened to responses that failed to parse or were truncated — a reader cannot rule out that some runs quietly dropped out of the denominators. The Docker image is cited as jyxsu6/medagentbench with no tag or digest, the HAPI FHIR JPA version is not given, and the AgentBench fork (Section 2.4) has no commit pinned.

Cost is discussed but never reported. Section 3 says the 300-task, 100-patient size was chosen "to achieve accurate estimates of performance at reasonable prices", yet there is no dollar figure, no token count, no per-model runtime, and no evidence offered for "accurate estimates" since no interval is computed anywhere in the paper. Same for the 8-round cap (Section 2.4.3): the fraction of failures that were cap exhaustion versus wrong answers is never given, which leaves Claude's Hard SR of 23.33% (Table 4) ambiguous between a reasoning limit and a budget limit.

One more that a benchmark paper should address head-on and this one does not: contamination. The 300 tasks and their reference answers are on public GitHub. Any model trained after release may have ingested them, which directly undermines the stated purpose of tracking progress over time. A held-out split, or at least a stated policy and a canary string, belongs in the paper.

A note on the version I received, per the rules I work under: this is the plain-text export of the Qeios page. Tables 1 through 4 came through as tab-separated rows and are legible, and I have relied on them.

Formatting did not survive — Table 4's caption says "The highest success rate in each column is highlighted in bold", and no bolding is present in my copy, so I could not verify which cells the authors marked. Figures 1, 2, and 3 are present only as captions, so my comments on the error taxonomy rest on the Section 2.5.2 text rather than on the figure itself.

Specific points

1. **Section 2.4 and 2.4.3 — POST requests are never executed against the environment.** Only GETs reach the server; POSTs get a JSON-loadable check and an unconditional success signal. Half the benchmark (150 action tasks, Section 2.5.1) is therefore graded on payload strings, and the Action SR column of Table 3 measures payload formatting plus rule coverage, not environment outcomes. Please either execute writes against a snapshot-restored server or state plainly in the Abstract and Section 2.4 that action tasks are graded offline.
2. **Section 2.4.1 — the graders are unvalidated.** "The grader and reference solution for each task category is manually curated", with "many rule-based sanity checks" for actions, and nothing on how those rules were validated. For a benchmark that will be cited as ground truth, please report an audit: sample a fixed number of scored-as-failure trajectories per model, have a clinician not involved in writing the rules adjudicate, and report the false-failure rate with agreement statistics. Semantically valid but non-matching FHIR payloads are the specific risk.
3. **Table 3, Action SR column versus Abstract — the headline inverts the paper's own safety framing.** Claude 3.5 Sonnet v2 leads Overall (69.67%) and Query (85.33%) but sits fifth on Action SR at 54.00%, behind Gemini-1.5 Pro 71.33%, Qwen2.5 64.00%, GPT-4o 56.00%, and DeepSeek-V3 54.67%. Report the split in the Abstract rather than a single averaged number.
4. **Table 3 — single-run numbers with no uncertainty, presented as a ranking.** Add repeated runs (at a minimum of three per model), report intervals, and use McNemar on the paired per-task outcomes for pairwise claims. Note that temperature zero does not guarantee determinism for API-served and MoE models, so repeats are informative even at temperature 0. Until then, please avoid "the best model" language for differences under about 8 points.
5. **Abstract, Section 1, Section 2.5 — "significant variation in performance across task categories" is unsupported in this version.** No per-category table exists; only query/action (Section 2.5.1) and difficulty (Table 4) breakdowns are reported. Add a success-rate-by-category table for all 10 categories, or drop the claim. Also, replace "significant" with either a test or a neutral word.

6. **Section 2.4.3 and 2.5.2 — format handling is applied unevenly across vendors.** Gemini's `tool_code` separators are stripped by the harness; no equivalent leniency is described for anyone else, while Gemini 2.0 Flash still fails 54% of actions on format, and Gemini 2.0 Pro lands at 18.00% overall (Table 3). Please report, per model, the split between format failures and content failures, and add an ablation using each provider's native function-calling API. Without it, readers cannot separate agent capability from prompt-protocol fit — and the benchmark's stated purpose is to measure the former.
7. **Appendix A.3, Table 4 — difficulty is non-monotonic for two models, and the group sizes are never stated.** Gemini-1.5 Pro: Easy 82.22%, Medium 45.83%, Hard 63.33%. Llama 3.3: Medium 38.33% below Hard 46.67%. Separately, the reported percentages are only consistent with group sizes of 90 easy, 120 medium, and 90 hard (for instance, Gemini 2.0 Flash Easy SR 98.89% = 89/90; Gemini-1.5 Pro Medium SR 45.83% = 55/120), and every row's Overall SR reproduces exactly under that weighting — but the paper never states those counts. Please state them and address the non-monotonicity.
8. **Section 2.4.2 — no model snapshots, no o3-mini reasoning effort, no serving precision for open-weights models.** Give dated model identifiers, the evaluation window, the o3-mini reasoning-effort value, and the Together AI serving precision or quantization. The "gap between closed and open-weights LLMs" claim in Section 2.5 cannot be evaluated without the last of these.
9. **Section 2.1 — task authorship and independence.** The 300 tasks were written by two of the authors (KB, JHC), both internal medicine. There is no second-rater review of clinical validity and no reporting of how many distinct templates the 300 items instantiate; Table 1's placeholders suggest substantial templating. Please report template counts and add an independent clinician review of a sample, ideally from outside Stanford given the single-site cohort.
10. **Section 2.5 versus Section 1 and Table 3 — model count mismatch.** Section 2.5 opens with "The performance of 11 state-of-the-art LLMs", while Section 1 says 12 and lists 12, and Table 3 contains 12 rows. Fix the "11".
11. **Section 2.1 versus Table 1 — task-type lists do not match.** The prose lists eight types including "patient communication" and "documentation"; Table 1 shows seven broad categories, and neither of those two appears. The paper also refers throughout to 10 specific categories that are never enumerated. Please give the full list of 10 and reconcile it with Table 1's grouping.
12. **Contamination is not discussed.** Tasks, reference logic, and environment are public. For a benchmark meant to "track progress", state the contamination policy: a held-out split, canary

strings, or at minimum an explicit caveat that post-release models may have seen the items.

13. **Section 2.2 versus Section 2.2.1 — date framing is confusing.** Section 2.2 says data covers "the last five years (November 13, 2018 as the cutoff date)", while Section 2.2.1 anchors the cohort on a sodium lab drawn the morning of November 13, 2023, and Table 1's contexts fix "now" at 2023-11-13T10:15:00+00:00. November 13, 2018, is the start of the window, not a cutoff. Please also say whether the fixed "now" is applied before or after the patient-level timestamp jittering described in Section 2.2, since "within last 24 hours" tasks depend on that ordering.
14. **Section 2.4.3 and Section 3 — interaction budget is never characterized.** With an 8-round cap, report the distribution of rounds used, the share of failures caused by cap exhaustion, and per-model token and dollar cost. Section 3 justifies the benchmark size on cost grounds without reporting any cost.
15. **Section 2.5.2 — the error analysis is one number for one model.** The only quantity given is Gemini 2.0 Flash at 54% invalid actions. A table of error categories by count and model would carry most of the paper's diagnostic value and is cheap to produce from saved trajectories.
16. **Section 2.4.1 — the pass@1 justification is sound but incomplete.** Single-attempt evaluation mirrors deployment, agreed. But deployment reliability is a variance question, and variance is exactly what a single run cannot report. Section 3 defers repeat-consistency to future work; it belongs in this paper, at least as a small study on the action subset.
17. **Minor, Section 2.5.2 —** the quoted erroneous output `"[\"value\": 5.4]"` is not valid JSON in any form, which makes the example hard to read. If the model emitted `{ \"value\" : 5.4 }`, please correct it.
18. **Minor, Section 1 —** the "first benchmark requiring autonomous interactions with medical records environments" claim is appropriately hedged, but AgentClinic (reference 21) is also an interactive clinical environment. One sentence distinguishing record-level FHIR interaction from simulated patient dialogue would preempt the objection.

Recommendation

Major revisions. The artifact — clinician-written tasks, a FHIR-compliant environment, public code, and image — is a genuine contribution, and I want to see it published. But the quantitative layer needs rebuilding before the numbers can be cited: writes are never executed against the environment on half the tasks, the graders have no reported validation, the leaderboard rests on single unreplicated runs with no intervals and no test, format handling favors one vendor, and the Abstract's claim about variation

across task categories has no table behind it. Points 1 through 6 require new work rather than rewording; the rest is reporting that the authors almost certainly already have on disk.

Declarations

Potential competing interests: No potential competing interests to declare.