

Reading Carryover Across Patients in a Clinical Agent

A two-model study of a diagnosis agent working eight patients per session on the Synthetic Hospital benchmark

Embedded Risk Analytics · A Fathom Case Study · September 2026

Synthetic Hospital, published in September 2026 by Park, Chen and Dettmers, gives AI agents a simulated electronic health record to work in. It holds 1,268 synthetic patients, each with a longitudinal chart of encounters, notes, results and documented history, served through a simulation of a hospital record system built on the FHIR standard. The benchmark splits the patients into 200 public, 268 held-out and 800 training cases and scores agents on four clinical tasks, diagnosis, summarization, evidence retrieval and imaging indication, against reference labels the agent never sees.

This study uses the diagnosis task. For each patient, an agent opens the chart, reads encounters, notes and results through 13 record tools such as chart search and encounter detail, and submits the patient's active diagnoses and chronic conditions, each with an ICD-10 code. The agent has 40 actions per patient. The benchmark scores each submission with a severity-weighted F1, a figure between 0 and 1 that rewards finding every true diagnosis (recall), penalizes listing diagnoses the patient does not have (precision), and gives serious conditions more weight. We ran the agent loop that ships with the benchmark through its environment interface, which keeps the coded problem list and the current visit's assessment out of the agent's reach, so the agent has to find the evidence in the chart.

We ran the task the way production agents tend to run, with several patients handled in one session, on two model families. The stock agent carried diagnoses from earlier patients into later patients' lists, with nothing in the later chart to support them. With Fathom's repair in front of the step where the agent submits, those copied diagnoses fell to zero on GPT 5.6 Luna and to one on DeepSeek V3.2, and the benchmark score rose by 0.06 to 0.10 on both models while recall held or improved.

Fathom checks what an agent commits against the record it has built.

Fathom, the hosted service from Embedded Risk Analytics, reads an agent's own actions and tool returns and reconstructs its committed state, the set of facts the agent has actually established from its record. The read reports where the agent later acts against that state. The repair runs in front of a single step of the agent's loop, evaluates the action the agent proposes against the committed state, and returns proceed, filter or reground. Proceed lets the step run as proposed, filter keeps the parts the record supports, and reground returns the conflicting facts to the agent so it can try again. Fathom works from the record a run already produces, needs no access to model internals, and runs on any model.

We asked the agent to work a queue of patients in a single session.

Agents that code charts, review claims or prepare prior authorizations rarely handle one case at a time. They work a queue, and the conversation that carries each case also carries the cases before it. We reproduced that arrangement. The agent worked the first 200 public patients in order, eight to a session and 25 sessions in all, with its own action budget and its own score for every patient. We ran two model families through OpenRouter, GPT 5.6 Luna from OpenAI at low reasoning effort and DeepSeek V3.2 from DeepSeek at the benchmark's registered settings. Each ran first as shipped, which we call the stock agent, and then with Fathom's repair in front of the step where the agent submits its diagnoses.

A queue places earlier patients inside the agent's working record.

In a single-patient session, everything the agent holds concerns one patient. In a queue, the conversation carries the agent's reads, reasoning and submissions for every earlier patient, and the loop keeps them in view, the recent turns in full and older turns in shortened form. By the eighth patient, earlier patients filled roughly 3,700 characters of what GPT 5.6 Luna saw at the moment it submitted. A diagnosis the agent committed for one patient can then resurface in the next patient's list, sourced from the agent's own history. We call the cost an agent pays to stay consistent with its own history reflexive burden, and the queue turns this form of it into a count.

A carried diagnosis fails its own patient and repeats an earlier patient's entry.

A submitted diagnosis counts as carried over when it matches none of the current patient's labels, rests on nothing the agent read for that patient, and repeats by name a diagnosis the agent submitted for an earlier patient in the same session. Neighboring patients in the benchmark often share clinical themes, so some repeats happen by coincidence. To measure that floor, we took single-patient runs of the same model on the same patients, grouped them into the same eight-patient sets, and applied the same test. Those sessions never saw one another, so every repeat they show reflects chance.

The stock agent carried diagnoses from patient to patient.

Across two runs of GPT 5.6 Luna, the stock agent carried 18 and 12 diagnoses into the 175 patients that followed another in their session, against about 8 expected by chance. Copies such as "34 weeks gestation of pregnancy" and "History of cesarean delivery" traveled from the first patient of a session to the fifth. In a handful of episodes, the agent also called a chart tool with an earlier patient's identifier. DeepSeek V3.2 carried 14 diagnoses in its stock run. The excess over chance stays small, roughly 2 to 6 percent of later patients on GPT 5.6 Luna, and each instance writes one patient's history onto another patient's problem list.

The benchmark score alone would miss the failure.

A carried diagnosis costs a patient one false item among about seven, and later patients in a session scored within run-to-run variation of the first patient in each session. A team watching the score would

see a steady agent. A read of each submission against the current patient's own record separates a diagnosis the chart supports from one the agent brought in from its history.

The repair removed the copied diagnoses and lifted the score.

The repair ran in front of the submission step, where it checked each proposed diagnosis against the committed state built from the current patient's record. We changed nothing else, at the same model, the same patients and the same session grouping. Table 1 sets the stock agent and the repaired agent side by side.

Model and run	Agent	Score	Precision / recall	Carried diagnoses	\$ per patient
GPT 5.6 Luna, run 1	Stock	0.584	0.54 / 0.73	18	0.0090
	With the repair	0.688	0.70 / 0.73	0	0.0085
GPT 5.6 Luna, run 2	Stock	0.620	0.57 / 0.75	12	0.0095
	With the repair	0.677	0.70 / 0.71	0	0.0088
DeepSeek V3.2	Stock	0.545	0.52 / 0.63	14	0.035
	With the repair, fitted setting	0.643	0.64 / 0.68	1	0.046

Table 1. Eight-patient sessions over the same 200 public patients. Score gives the benchmark's severity-weighted F1 averaged over patients. Carried diagnoses count across the 175 patients that followed another in their session. Spend gives OpenRouter's billed cost per patient.

On GPT 5.6 Luna, the repair raised the score by 0.103 in the first run and 0.057 in the second (95 percent intervals 0.067 to 0.141 and 0.026 to 0.089), lifted precision from about 0.55 to 0.70 at the same recall, and cost the same or slightly less per patient. The first stock run drew a low baseline, so the second run gives the better single estimate of the gain.

DeepSeek V3.2 reproduced the result.

DeepSeek V3.2 works a chart differently. It made about 33 chart-tool calls per patient against about seven for GPT 5.6 Luna, and it re-read the same material far more often. With the repair at its default setting, the score rose from 0.545 to 0.612, precision rose from 0.52 to 0.61, and recall rose from 0.63 to 0.67. With a setting fitted to DeepSeek's reading pattern, the score reached 0.643, a gain of 0.098 over stock (95 percent interval 0.061 to 0.137), and carried diagnoses fell from 14 to 1.

The right configuration depends on the model.

At its default setting on DeepSeek V3.2, the repair pushed ten patients to a forced submission at the end of their action budget, and the agent's calls naming another patient rose from 47 to 67. The fitted setting brought those forced submissions down to two and those calls down to 36 while keeping the full score gain. How often a model re-reads its own material points to the setting that fits, and Fathom reads that

pattern from the same action stream it already consumes. On DeepSeek V3.2, both configurations cost about 30 percent more per patient than stock while using the same number of calls and input tokens, and the billing records do not yet show why. On GPT 5.6 Luna, spend held level with stock.

A committed-state repair protects each case in a clinical workflow.

Coding, prior-authorization and chart-review agents work patients in batches, and every batch hands the agent a growing record of other people's care. A diagnosis that crosses from one patient to another carries clinical and privacy weight well beyond its effect on any score. Fathom reads each commitment against the record of the case in front of the agent, from the action stream the run already produces, with no access to model internals and on any model. The same read that removes a carried diagnosis also tells a team which configuration a given model needs before the agent reaches production.

Method and conditions.

The Synthetic Hospital benchmark (arXiv 2609.30027) supplied the charts, the labels, the scorer and the agent loop, run through its environment interface. The agent worked the first 200 public patients in ascending order, grouped into 25 sessions of eight, with its own environment episode and 40-action budget for each patient. Both models ran through OpenRouter on 27 and 28 September 2026. GPT 5.6 Luna ran twice per condition and DeepSeek V3.2 once per configuration. We registered each test before the run and compared conditions with a paired bootstrap over patients at 10,000 resamples. One DeepSeek V3.2 episode ended on a network reset, which left 199 patients in that run. The benchmark and its patients are synthetic, and nothing in this study applies to clinical care.