CLINICAL AI AUDIT

Ada Health Symptom Checker

An adversarial clinical audit evaluating diagnostic reasoning, urgency logic, and questioning behavior across 11 structured test cases.

Auditor: Dr. Rameesha Chohan, MBBS, FCPS 1
Methodology: Unsolicited adversarial audit — manual protocol, 11 cases across 5 failure modes
Date: April 2026
Status: *Portfolio artifact — independent, unsolicited, not commissioned by Ada Health*

Executive Summary

Ada Health's symptom checker was subjected to an 11-case adversarial audit designed to probe the boundary between pattern recognition and clinical reasoning. Cases were constructed across five structural failure modes and run manually using the mobile app, with outputs logged to a structured MySQL database and analyzed in Python.

The audit identified a categorical performance boundary: Ada performs reliably on classic, single-system presentations requiring pattern recognition alone, and fails systematically when cases require comorbidity integration, geographic context, or logical questioning sequences. This is not a marginal performance gap — it is a structural reasoning limitation.

45.5%	9.2/10	45.5%	0/5
of cases RED zone	avg. RED severity	top Dx accuracy	RED danger Dx ranked

Central finding: Ada is a competent pattern matcher but a poor clinical reasoner. It performs at the symptom surface but fails at the reasoning layer — and the cases where it fails carry the highest clinical stakes.

Audit Methodology

Design Principles

This audit was designed to be adversarial rather than representative. Cases were not selected to reflect average Ada usage — they were engineered to probe specific reasoning boundaries. A symptom checker that passes an adversarial audit at this level has earned its clinical credibility. One that fails has identified the exact conditions under which it cannot be trusted.

Case Construction

- 11 cases run across 5 pre-defined failure mode categories
- Cases entered as a patient would, not as a clinician — no volunteered information
- Only information explicitly requested by Ada was provided
- Cases 4 and 5 run as a paired comparison to test severity escalation recalibration
- All outputs logged to a structured MySQL database (audit schema, 3 tables)
- Analysis performed in Python using pandas and matplotlib

Scoring Framework

Each case was scored across four binary dimensions and assigned a failure zone:

Metric	Definition	Scale
urgency_correct	Did Ada assign the clinically appropriate urgency level?	Binary (0/1)
top_dx_correct	Was the #1 ranked diagnosis the most appropriate given the presentation?	Binary (0/1)
danger_dx_ranked	Was the most dangerous plausible diagnosis present in the differential?	Binary (0/1)
severity_score	Clinical stakes of what was missed or mishandled	1–10

Results

Zone Distribution

Of the 11 cases run, 5 were classified RED (45.5%), 2 YELLOW (18.2%), and 4 GREEN (36.4%). RED cases carried an average severity score of 9.2/10 — meaning the majority of Ada's failures occurred on the presentations with the highest clinical stakes.

Zone	Cases	% of Total	Avg. Severity	Max Severity	Clinical Stakes
RED	5	45.5%	9.2/10	10/10	Critical
YELLOW	2	18.2%	5.0/10	5/10	Moderate
GREEN	4	36.4%	1.5/10	2/10	Low

Scoring Metrics

Ada achieved 81.8% urgency accuracy across all cases. However, urgency accuracy is the weakest test of a clinical AI system: it measures whether an alarm fires, not whether the reasoning behind it is correct. The more clinically meaningful metrics tell a different story:

- Top diagnosis correct: 45.5% (5/11)
- Danger diagnosis ranked in differential: 45.5% (5/11)
- Cases where urgency was correct but diagnosis was wrong: 2 cases — the urgency-diagnosis incoherence pattern

Key insight: A patient who receives the right alarm with the wrong diagnosis will underestimate what is happening to them. Urgency accuracy without diagnostic accuracy is not safety — it is a false sense of safety.

Analysis Chart

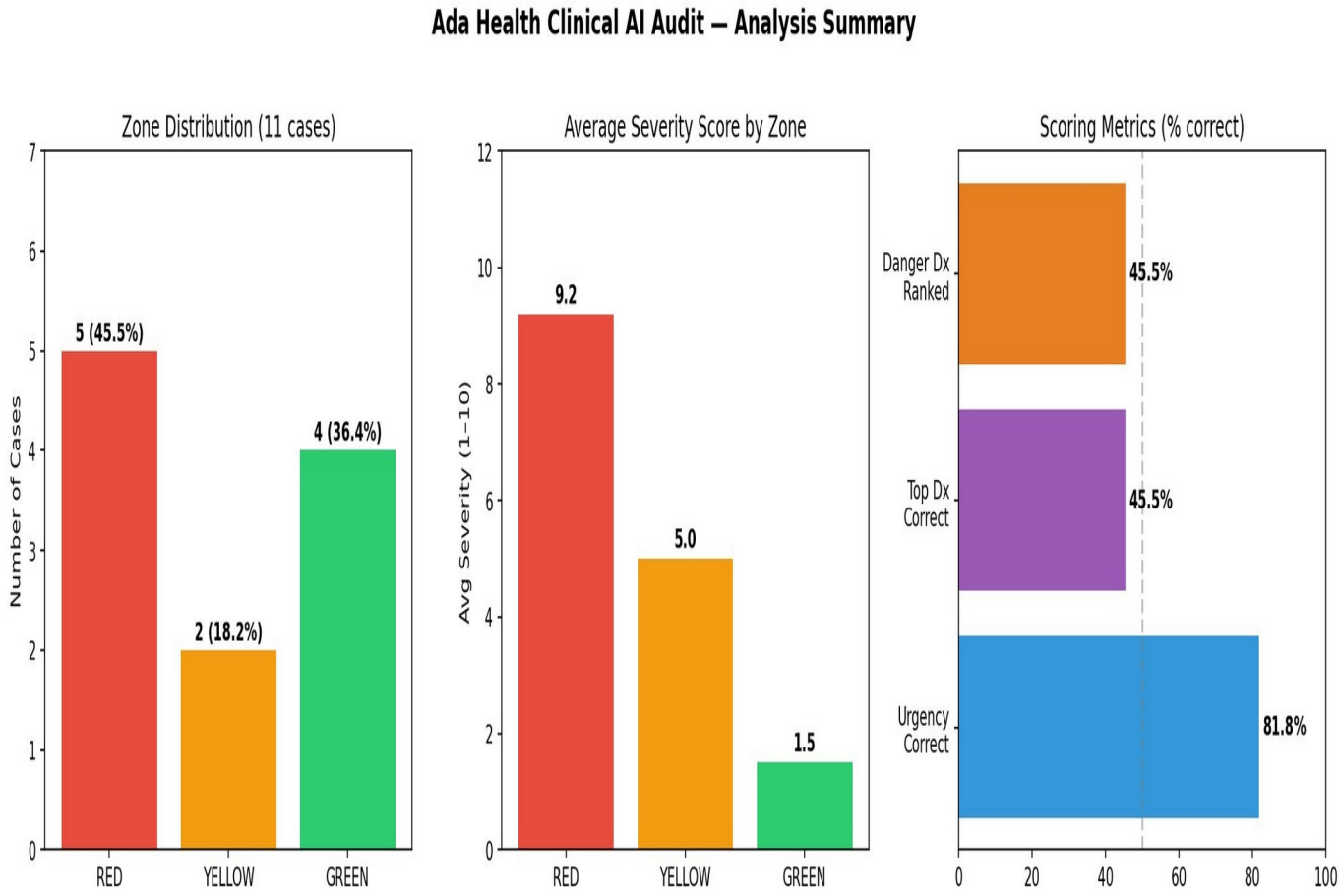

Figure 1. Zone distribution, average severity by zone, and scoring metrics across 11 audit cases.

Structural Findings

The audit identified five structural failure modes — patterns that appear repeatedly across cases and reflect design-level limitations rather than isolated errors.

Failure Mode	Cases	Description	Type
Comorbidity integration failure	Cases 4, 5, 7, 8	HTN/Diabetes confirmed but never interrogated further	Process
Questioning logic sequencing	Cases 1, 6, 9	Questions asked out of logical order relative to confirmed variables	Process
Geographic blindness	Case 3	No location data collected — entire output calibrated to Western disease priors	Design
Urgency-diagnosis incoherence	Case 5	Correct urgency label paired with wrong #1 diagnosis — patient misdirection	Safety
Symptom severity non-recalibration	Cases 4→5	Differential unchanged between blurred and complete vision loss	Reasoning

Finding 1 — Comorbidity Integration Failure

> **Pattern:** Ada collects confirmed co-morbidities as static checkboxes and never interrogates them further.

This pattern appeared in four consecutive cases. In each, a clinically significant comorbidity was confirmed early in the assessment and then treated as background noise for the remainder of the interaction:

- **HTN confirmed, blood pressure reading never requested, hypertensive retinopathy absent from differential** Case 4 (HTN + visual symptoms)
- **HTN confirmed, BP never asked, CRAO absent — a stroke equivalent with a 90-minute treatment window** Case 5 (HTN + sudden vision loss)
- **Diabetes confirmed, zero diabetic follow-up questions asked, diabetic neuropathy ranked #3 at 3% in its textbook presentation** Case 7 (Diabetes + foot tingling)
- **HTN confirmed, BP never asked — process failure noted, outcome unaffected only because the presentation was unambiguous without it** Case 8 (HTN + thunderclap headache)

The consistent pattern across four cases suggests this is not an edge case — it is how Ada's questioning engine is designed. Co-morbidities are collected for differential generation but do not feed back into the questioning logic.

Finding 2 — Questioning Logic Sequencing Failures

Pattern: Ada asks downstream questions before upstream variables have been established.

Clinical reasoning follows a logical dependency structure: you cannot ask meaningful questions about a variable until the parent variable is confirmed. Ada violated this structure in three cases:

- **Head injury confirmed — mechanism of injury, loss of consciousness, and injury severity never asked. These are the three discriminating questions between minor head injury and intracranial hemorrhage.** Case 1
- **Pregnancy confirmed — Ada then asked about recent unprotected sexual intercourse. This question is relevant when pregnancy is uncertain; it is the wrong question for a confirmed pregnant patient.** Case 6 (Vaginal bleeding in pregnancy)
- **Pregnancy not yet confirmed — Ada asked about gestational age at 23 and 37 weeks before establishing whether the patient was pregnant at all.** Case 9 (RLQ pain, uncertain pregnancy)

Cases 6 and 9 represent the same logic error in opposite directions — Ada's pregnancy questioning module appears to have a fixed sequence that does not adapt to the confirmed state of the upstream variable.

Finding 3 — Geographic Blindness

Pattern: Ada collects no location data at any point. The entire output is implicitly calibrated to a Western patient population.

Case 3 presented a patient with episodic fever, chills, and myalgia in the context of a local malaria outbreak. Ada's differential: tick-borne relapsing fever (#1, 40%), familial Mediterranean fever (#2, 20%), AML (#3, 6%), ALL (#4, 5%).

Malaria was absent entirely. Ada asked about tick exposure and Mediterranean descent — questions designed to unlock rare diagnoses prevalent in Western patient populations — while never asking about location, recent travel, or local disease patterns.

Scope of impact: This is not a single-case failure. Geographic blindness affects every non-Western user of Ada in every session. The tool's prior probabilities are anchored to a population that does not reflect the majority of the world's disease burden.

Finding 4 — Urgency-Diagnosis Incoherence

Pattern: Correct urgency label paired with an incorrect #1 diagnosis produces patient misdirection even when the alarm fires.

Case 5 (HTN + sudden complete monocular vision loss) demonstrates a failure mode that does not appear in standard AI evaluation frameworks. Ada correctly triggered an emergency alert — but ranked Migraine as the #1 diagnosis at 40%.

A patient reading 'Migraine — seek emergency care' will calibrate their urgency response to the diagnosis, not the label. They will assume the emergency instruction is precautionary. Central retinal artery occlusion — a retinal stroke with a 90-minute treatment window — was absent from the differential entirely.

This case also demonstrated that Ada's differential did not materially change between Case 4 (blurred vision) and Case 5 (complete vision loss in the same eye) — indicating that symptom severity escalation fails to trigger diagnostic recalibration.

The Performance Boundary

The most analytically significant finding from this audit is not any individual case failure — it is the categorical boundary between Ada's GREEN and RED performance zones.

GREEN Cases (4/4 correct)	RED Cases (0/5 correct)
✓ Classic symptom clusters	✗ **At least one of:**
✓ Single organ system	✗ Confirmed comorbidity needing integration
✓ No comorbidity integration required	✗ Geographic context dependency
✓ No geographic context needed	✗ Questioning logic sequencing requirement
✓ No questioning sequence logic required	✗ Symptom severity recalibration needed

This is not a performance correlation — it is a categorical split. Every GREEN case passes all four scoring dimensions. Every RED case fails top_dx_correct and danger_dx_ranked entirely (0/5). The boundary predicts outcome with 100% accuracy within this dataset.

Implication for deployment: Ada can be trusted to handle textbook presentations in populations that match its training priors. It cannot be trusted when patient complexity exceeds the pattern recognition layer — which is precisely when accurate AI triage matters most.

A Note on Audit Methodology

This audit was conducted using an unsolicited adversarial framework — designed, executed, and analyzed independently with no access to Ada's internal architecture, training data, or engineering documentation. All conclusions are based solely on observable input-output behavior.

The value of this methodology is precisely its externality. It replicates the experience of the patient — who also has no access to the system's internals and must rely entirely on what Ada outputs. The audit asks the same question a patient implicitly asks every time they use the tool: can I trust this?

What This Audit Is Not

- This is not a regulatory submission or a clinical validation study
- It is not a complete evaluation of Ada's performance across its full user base
- It is not a claim that Ada is unsafe for all use cases

What This Audit Is

- A structured, reproducible methodology for identifying reasoning-layer failure modes in clinical AI tools
- A demonstration that adversarial auditing surfaces failures that standard benchmark testing misses
- A portfolio artifact demonstrating clinical AI audit capability — available for adaptation to any symptom checker, triage tool, or diagnostic AI system

About the Auditor

Dr. Rameesha Chohan, MBBS, FCPS 1

Clinical AI Auditor & Health Tech Consultant

Dr. Chohan is a physician-turned-clinical-AI specialist with formal medical training and a focused practice in evaluating AI-assisted clinical tools. She works with health tech startups to identify where clinical AI logic breaks, bridging the gap between engineering teams and practicing clinicians.

Her audit methodology combines adversarial case design, structured SQL-based output logging, and Python-driven statistical analysis to produce reproducible, evidence-grounded findings that engineering and product teams can act on.

This case study is an independent portfolio artifact. It was not commissioned by Ada Health and reflects no affiliation with or access to Ada Health's internal systems. All audit inputs and outputs are documented and available on request.