A Deployed Framework for Structured Clinical Extraction from Multi-Turn Health Conversations: From Model Selection to Scenario-Based Release Gating

Krishnam Gupta†, David Mbatuegwu, Stanslaus Mwongela, Yohann Richard and Rouella Mendonca

Audere, Seattle, WA, USA

†*Corresponding author. E-mail: krishnam@auderenow.org*

Background. Conversational health agents in low- and middle-income country settings generate clinically important information as unstructured, multi-turn dialogue. Converting them into structured records is difficult: extraction is stochastic, disclosures may be corrected across turns, uncertain models can silently omit fields, and unsafe inference can create false clinical facts. Aggregate benchmarks identify promising models but do not establish release stability or characterize field-level risk.

Methods. We describe a structured-extraction framework deployed across four HIV programmes in South Africa and Zimbabwe (one also covering TB) that together have served more than 170,000 users. An LLM extracts canonical values into programme-specific schemas drawn from 137 declared fields; deterministic code derives fixed clinical mappings and field-level validation preserves unaffected outputs. A benchmark of 398 counselor- and clinician-annotated conversations selects candidate models. Before release, 134 scenario-based tests across 14 clinical categories repeatedly evaluate complete production configurations. Structured-record assertions are checked deterministically, while conversation-dependent behavior is evaluated by an operational LLM reporter that is itself audited.

Results. The selected primary and incumbent were nearly tied on the aggregate benchmark, but critical-field analysis exposed an HIV-status omission deficit. A within-sample rerun under a later configuration changed the critical-field model ordering, the incumbent regressing while the primary remained stable; the aggregate score again concealed the change. Across 356 matched scenario-programme cells, the primary passed 8.4 percentage points more cells than the deployed cross-provider fallback. The fallback also produced substantially more fabrications, including 20 unsupported HIV-status assertions against one. Removing both static guidance layers reduced stable-pass rate from 87.3% to 63.9%; layer-separation experiments showed that guardrails and field descriptions provide overlapping protection. Auditing the reporter against direct record checks exposed defects in the test specifications, the evidence supplied to the reporter, the comparison rules and the reporter itself; correcting the gate left the model ordering and failover conclusions unchanged.

Conclusion. Reliable deployment required more than model selection. It required layered safeguards, repeated configuration-level release gating, analysis of error composition, and validation of the gate itself. Assertions encoded in the structured record should be checked deterministically, with LLM judgment reserved for conversation-dependent behavior.

Keywords: Structured clinical data extraction; large language models; scenario-based testing; release gating; clinical AI deployment.

© 2026 The Authors. Open Access chapter published by World Scientific Publishing Company and distributed under the terms of the Creative Commons Attribution Non-Commercial (CC BY-NC) 4.0 License.

1. Introduction

Conversational AI health agents are moving from pilots into sustained deployment, particularly on WhatsApp in LMIC settings (for example the Shesha TB chatbot[1]). Our system operates across four HIV programmes in South Africa and Zimbabwe (one also covering TB). Together they had served more than 170,000 users in more than 250,000 multi-turn conversations by September 2026. Clients disclose HIV test results, missed ART refills, new symptoms and safety concerns; none of it arrives as structured data. It arrives as chat, which the system turns into a structured clinical record under a programme-specific schema drawn from 137 declared fields. Providers and programme workflows act on this record rather than the chat itself (Section 3), so an extraction error can become a missed or unnecessary clinical follow-up.

Scope. The deployed product also generates conversational replies and clinical summaries through separate LLM calls; *this paper concerns only the structured data-extraction component.* The schemas span HIV, TB, sexual and reproductive health, and mental health. The evaluation is HIV-dominant, and TB fields appear in the scenario suite but are not analyzed separately.

At first glance this is a structured information-extraction task, but in deployment several failure modes make it substantially harder than generic extraction benchmarks suggest. **Failures are silent**: an uncertain model may omit a disclosure without raising an error, and a dropped HIV disclosure is indistinguishable downstream from a topic that never came up. **Errors are clinically asymmetric**: missing a field and fabricating an HIV status do not carry the same consequence, and a well-formed wrong value is indistinguishable from a correct one without ground truth. **The pipeline is stochastic and multi-turn**: disclosures may be corrected or qualified across turns, relative dates resolve against conversation time, and assistant responses shape subsequent disclosures. Changing providers introduces further structured-output and validation failure modes (Sections 5.2, 5.3).

The gap we address is therefore a **testing and measurement gap**: how does a small team gate each release of a safety-critical, *stochastic, multi-turn* extraction pipeline, when the default instruments (aggregate accuracy, single test runs, LLM-based judging) can each fail silently? Our central operational distinction is between model selection and configuration-level release gating: the benchmark identifies viable candidates, while repeated scenario testing establishes release stability and characterizes field-level risk, which then informs fallback and monitoring policy.

Contributions. We establish three claims from a deployment across four HIV programmes. **(C1) Selection and release gating serve different functions and require different instruments.** An aggregate benchmark ranks candidates but cannot establish release stability or characterize field-level risk; repeated scenario testing can (Sections 5.1, 5.3). **(C2) Fallback policy must consider error composition, not aggregate error rate alone.** Two models within 8.4 points on pass rate differ sevenfold in fabrication, and removing guidance disproportionately increases unsupported extraction where abstention is correct (Sections 5.3, 5.4). **(C3) A release gate must itself be audited.** Cross-checking the operational LLM reporter against deterministic record checks surfaced 88 reporter-pass/record-fail disagreements, revealing defects in the test specifications, the evidence supplied to the reporter, the comparison rules and the reporter, and motivated a deterministic-first gate (Section 5.5).

Supporting these, we describe the deployed architecture (Section 3) and show that a single execution can miss intermittent failures (Section 5.2). Safety-escalation performance is evaluated separately,[2] extracted indicators support programme intelligence,[3] and service outcomes are reported elsewhere.[4]

2. Related Work

Structured clinical extraction from text is a long-standing biomedical-NLP task.[5,6] LLMs extended it toward few- and zero-shot settings, though performance varies by task, field and evaluation design,[7,8] temporal and relational extraction is error-prone,[9] and open models can be competitive.[10] Tool use and function calling are well established,[11–13] constrained generation is supported by several decoding techniques,[14,15] and libraries such as Instructor provide schema-validated outputs in production.[16] Yet conformance is not correctness, and reliability degrades as schemas grow,[17–19] motivating our hybrid split (Sections 3.3, 3.4).

Medical-LLM evaluation has often relied on exam-style question-answering benchmarks,[20,21] on which models do better than on practice-oriented tasks;[22] newer benchmarks such as HealthBench score realistic multi-turn conversations against physician-authored rubrics,[23] but still evaluate response quality rather than the structured record. Shesha demonstrates deployed conversational health workflows in sub-Saharan Africa[1] but does not examine the structured extraction layer studied here. LLMs also propagate injected false findings,[24] and outputs are non-deterministic even at temperature zero,[25] so a single-shot result does not characterize run-to-run stability.[26]

Scenario-based behavioral testing builds on CheckList-style suites,[27] extended to compound AI systems with operational failures that aggregate benchmarks may miss.[28] LLM-based evaluators make such testing scalable,[29,30] but a checklist study finds they often miss known quality degradations,[31] so we treat the reporter as an auditable component (Section 5.5). Related clinical examples derive MS progression from notes with an extract-then-derive design[32] and benchmark conversational access to structured glucose data.[33] Outside medicine, dialogue state tracking[34] faces the same accumulation problem, where standard metrics can misstate multi-turn performance.[35] We know of no public benchmark or tool for schema-constrained extraction from multi-turn counseling chat, so we compare candidate models under identical scaffolding rather than against an external baseline. We combine these ideas in a deployed release process built on repeated execution, error-type decomposition, direct record checks and field-level risk analysis.

3. Methods

Deployment context. Each programme runs a WhatsApp AI companion within an existing community HIV service, with the implementing partner's providers in the loop. Aimee (Shout-It-Now), Self-Cav (National Department of Health) and Coach Mpilo (PSI, covering TB) run in South Africa, and VimbAI (CeSHHAR) in Zimbabwe. Clients chat freely. The companion answers questions and asks for the fields the programme needs, and the extracted record feeds provider dashboards, risk stratification, AI follow-up, and targeted outreach.[3]

3.1. *System overview and the dual-pass design*

During an active conversation the product issues four LLM calls per user message (summary, per-message (PM) extraction, function calling, response), and a separate end-of-conversation (EoC) extraction runs once after the conversation closes. Only the two extraction calls are in scope. Figure 1 shows the end-to-end dataflow. The two passes share 18 core guardrail blocks; the EoC prompt composes further field-specific blocks for fields absent from the smaller PM schema.

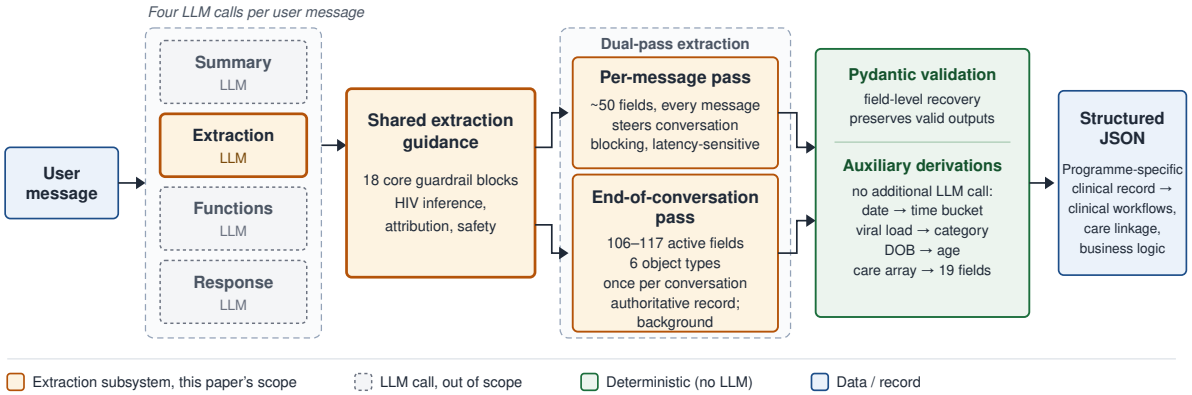


Fig. 1. End-to-end extraction dataflow (Section 3.1).

Active schemas vary by programme. In the releases evaluated here, the **PM pass** (~50 fields) runs on *every* user message; its outputs steer the live conversation and feed the Section 3.5 escalation, making it blocking and latency-sensitive. Each programme's active schema held 106 to 117 top-level fields drawn from 137 declared; counting nested custom-object attributes separately, 192 elements carry a field description. The **EoC pass** runs once with the full programme-specific schema and six custom object types on a lower-cost service tier, and serves as the authoritative record for downstream programme workflows. Within a conversation the final EoC record supersedes the PM record. The subsequent write into the longitudinal profile applies merge rules that prevent an `unknown` from overwriting a known value; the remaining reconciliation and provenance rules are in supplement Part 5, and alternatives to post-hoc validation (constrained decoding, fine-tuning, self-consistency voting) in Part 4.

Cost, latency, and query composition. The programmes run under a binding per-patient cost constraint and the per-message pass is synchronous, so its latency is user-visible: each pass is a single call returning all applicable fields under a stable, cacheable prefix, with inference-time reasoning disabled on both. Guidance sits at the most structural layer that can express it: schema constraints for representation, field descriptions for field-local semantics, and prompt guardrails for cross-field reasoning. All three stay static so the prefix caches; Section 5.4 examines the last two.

3.2. *Custom clinical objects, attribution, and guardrails*

Clinically relevant disclosures often describe *events* with internal structure, so we model six custom object types, each decomposing an event into independently validatable attributes while preserving subject attribution and multiplicity. We address known high-consequence

failure modes through 18 reusable core guardrail blocks (∼4K tokens) composed per prompt; the EoC prompt adds field-specific blocks for attribution, temporal and cross-field constraints not required by the smaller PM schema. Blocks follow the loop of Figure 2: a production or scenario failure yields a paired scenario test, the block is revised (grouping related rules, balancing positive and negative examples, separating overlapping concepts), and the configuration is re-gated. Every clinical object carries a `reported_for` attribution defaulting to the client, so a partner's or relative's HIV status never enters the client's record (supplement Part 4).

HIV-status inference, a worked example. Early models turned a generic "I tested positive" into `hiv_status: positive`. The block now populates HIV status only from an explicitly HIV-labeled result, a direct answer to the agent's own HIV-specific question, or narrowly whitelisted treatment or prevention evidence. Generic statements leave it `unknown` (supplement Part 4 gives the temporal and stopped-PrEP rules).

3.3. *Separating extraction from deterministic derivation*

Combining canonical extraction with clinical interpretation in one step created avoidable inconsistencies, such as a correct absolute date paired with a wrong time-bucket. We therefore separate the two: the LLM recovers **canonical values**, and deterministic code **maps** an absolute date to a time-period bucket, viral load to a category at fixed thresholds, date of birth to age, and the care-encounter array to the most recent date per care type.

One-to-nineteen care-linkage derivation. One extracted array becomes 19 flat care-linkage date fields, one per care type, reducing the independent omissions and contradictions that arise when all 19 are generated separately.

3.4. *Validation and graceful degradation*

The validation layer combines three mechanisms. **Schema exclusion** keeps derived fields out of the schema the LLM sees while retaining them in the stored output. **Selective requiredness** removes a field's default at schema generation, making it `required`: the model must emit either a clinical value or an explicit `unknown`, so an omission becomes a detectable validation failure rather than a silent absence. If the model still omits the field, recovery inserts `unknown`. That explicit `unknown` is not retained by the longitudinal-profile writer, so abstention must be measured before persistence. In production the flag is applied only to the safety fields.

Per-field graceful degradation ensures a single malformed field never discards an otherwise-good extraction: recovery removes or repairs only the malformed attributes, so a bad `onset_timing` costs that attribute rather than the record, and a failing array element is dropped rather than the array. This layer also runs normalizing validators before persistence; one such intervention is described in Section 5.2.

3.5. *Real-time safety escalation*

The deployed system includes a real-time, human-in-the-loop safety pathway: two extracted fields, self-harm and disclosure of sexual or gender-based violence, create a clinical-portal task for the implementing partner's providers, as do the platform's content-moderation flags.

Other extracted fields create routine follow-up tasks. In the Aimee pilot (preprint[36]), providers handled these tasks during weekday working hours under the partner's procedures, connecting clients to a nurse, social worker or external partner. Outside those hours the companion gave crisis information and hotline numbers. Tasks for concerning content were completed in a median of 14.0 hours, against 47.3 for routine follow-up. A separate evaluation of the layered safety architecture examines clinician escalation and downstream safety performance.[2] This paper evaluates the structured-extraction inputs that feed the pathway.

4. Evaluation Methodology

We use two evaluation tiers. A quarterly benchmark selects candidate models (Section 4.2). Roughly monthly, repeated scenario-based testing of the top two or three candidates gates each production configuration for release and identifies field-level weaknesses that inform fallback and monitoring policy (Section 4.1). Figure 2 shows how they connect to deployment and to the safety pathway of Section 3.5. The benchmark is scored deterministically against human labels. Scenario runs use two instruments. For the releases analyzed here, an operational LLM evaluator (**the reporter**) produced the gating verdict. We also rescored each test's unconditional record expectations **deterministically** from the extracted profile the test stores, with no LLM involved (cross-check in Section 5.5). Behavioral safety tests are reporter-scored and out of scope, so quantitative claims rest on the benchmark and the scenario evaluations below.

The selection tier scores the record alone, while the gating tier over-samples rare fields and also judges behavior conditioned on the record, because a wrong record changes what the companion says next. Each experiment states its scenario population (summary in supplement Table S3), and every inferential scenario comparison uses the repaired deterministic comparator of Section 5.5.

4.1. *Scenario-based testing, SBT (gating tier)*

SBT adapts scenario-based testing to stochastic, multi-turn clinical extraction through repeated execution, explicit expectations and stable-pass, flaky or stable-fail classification. Each test drives a multi-turn scenario through the production pipeline in each programme configuration it applies to (one test in one programme is a *cell*), using scripted utterances when exact wording is under test and an LLM simulator otherwise. The simulator and evaluation instruments receive the scenario's expected values; the extraction model does not. Unconditional record assertions (values the record must hold however the conversation unfolds) are checked deterministically, while conversation-dependent behavior is evaluated by the operational reporter (Section 5.5). Tests therefore include a prose behavioral specification and, where applicable, a structured expected record. The two were maintained independently during the evaluated releases, enabling separate record and behavioral checks but also permitting the specification drift examined in Section 5.5.

The matched suite contains 134 tests across 14 clinical categories: 120 data-extraction tests and 14 clinical or safety-response tests that also include structured-record assertions (taxonomy in supplement Part 2). Behavioral-only tests are reporter-scored and excluded from the deterministic comparisons. The suite is **expanded with production edge cases**

and rerun **for each release**, and is not used for the model-selection results of Section 5.1. Of the 134 matched tests, 84 predated the February 2026 model decision, 131 predated the July 2026 configuration decision, and three were written during the experiments reported here. No scenario set was held out (Section 7). The suite is enriched for ambiguous and adversarial disclosure, especially HIV-status scenarios, in which many safety-relevant positives are paired with a negative designed to invite over-extraction. **Scenario pass rates therefore describe an intentionally difficult, safety-enriched test distribution and are not estimates of production prevalence**; the real-conversation benchmark provides a more naturalistic reference distribution. Routine tests run twice and safety-critical tests run five times, and suite-wide rates use two runs unless stated otherwise.

4.2. *Annotated benchmark (selection tier)*

Dataset. 398 real counseling conversations from **two of the four deployed programmes**, Aimee (291) and Self-Cav (107), both in South Africa, collected from March to November 2025 (Aimee pilot) and from May 2025 to January 2026 (Self-Cav). They were annotated by trained HIV testing services counselors and clinicians, one annotator per conversation, so inter-annotator agreement could not be estimated (Section 7).

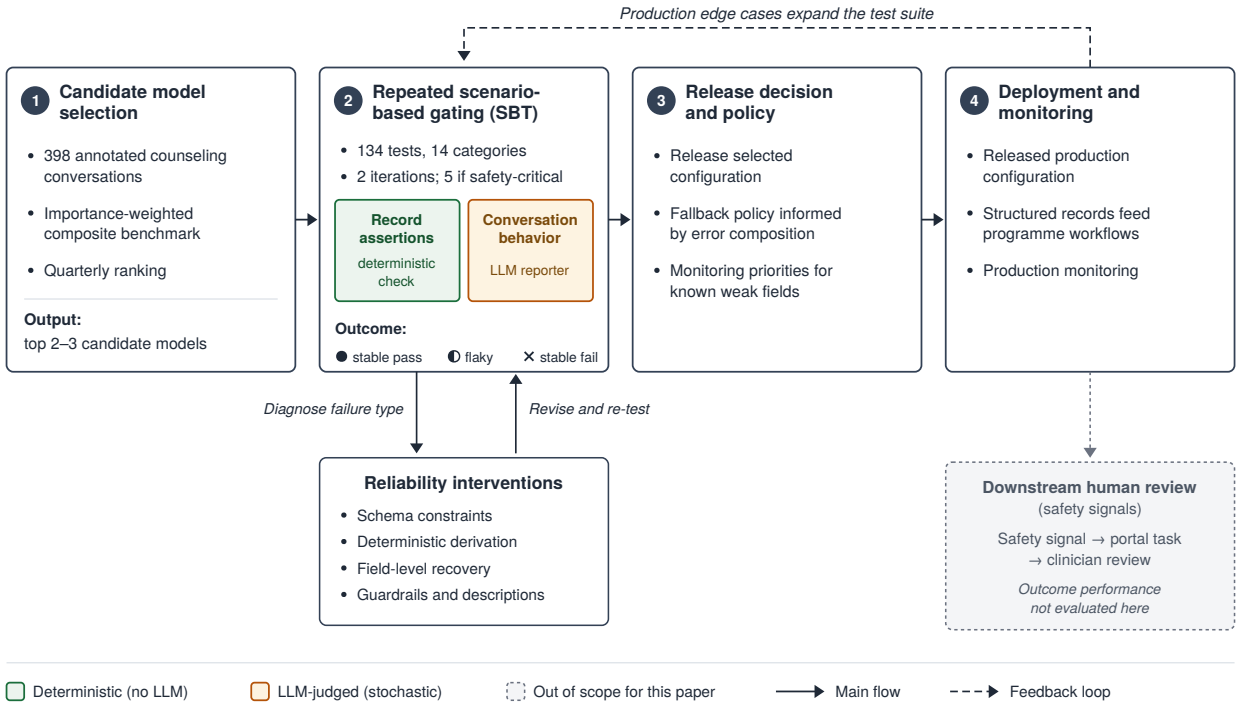


Fig. 2. Reliability loop from candidate selection to deployment. The quarterly benchmark ranks candidate models (Section 4.2); repeated scenario-based testing then gates each production configuration, checking unconditional record assertions deterministically before the operational reporter evaluates conversation-dependent behavior (Sections 4.1 and 5.5). The figure shows the corrected deterministic-first gate, with the historical gate audited in Section 5.5. Field-specific failover policy is supported by the evidence but is not yet deployed, and downstream human-review performance is not evaluated in this paper.

Metrics. Most fields are scored by exact match after normalization; list-valued fields use order-independent Jaccard similarity. The headline is an importance-weighted composite over critical and non-critical field accuracy, with penalties for false and missing extraction, discounted by a structured-output reliability term (specification and weights in supplement Part 1, held fixed across all models compared here, with the complete candidate-model comparison reported separately). Field-level errors are classified against the reference (the annotation, or a scenario's expected record): a *fabrication* asserts a value where the reference is `unknown`, an *omission* leaves a field empty where a value is expected, and a *substitution* emits a known value differing from the expected one. Pairwise comparisons use Wilcoxon signed-rank tests on paired per-conversation composites with conversation-level bootstrap intervals. Conversations with no returned record stay in every denominator as misses (Table 1: none for the primary, 5 of 398 for the incumbent). For the matched failover comparison of Section 5.3, where a test may contribute cells from more than one programme, we report test-clustered inference (a test-cluster bootstrap for intervals, a paired test-cluster permutation test for p-values); McNemar values elsewhere are cell-level and descriptive.

5. Results

5.1. *Aggregate parity concealed a field-level deficit*

The selected primary (GPT-5.2) and the incumbent (GPT-4o) differed negligibly on the aggregate composite (0.913 and 0.912; mean paired difference +0.0006 over 398 conversations, bootstrap interval including zero), so **the tie was broken on the evaluated cost profile**. The same cycle shortlisted the deployment chain evaluated throughout: the promoted primary (GPT-5.2), a cross-provider fallback (Claude Haiku 4.5) and an open-weight contingency (Qwen 3 Instruct), served versions in supplement Table S1. The fallbacks were selected for benchmark accuracy, provider diversity and cost that stays sustainable as programmes scale, rather than capability matching, so their suitability required separate scenario evaluation.

The aggregate tie concealed a clinically important difference (Table 1). Under the February configuration a Wilcoxon signed-rank test detected a distributional difference on critical fields favoring the incumbent ($p = 0.002$), driven by `hiv_status` omissions, although the mean paired difference was small and its bootstrap interval included zero. Both models were rerun on the same 398 conversations under the July configuration. Because the schema had roughly doubled, we repeated the comparison on a field set frozen to February: the primary-minus-incumbent critical-field difference was −0.0043 under February and +0.0123 under July (July 95% CI +0.0052 to +0.0202). The shift reflected mainly a decline in the incumbent's score, 0.884 to 0.872, while the primary rose slightly, 0.879 to 0.885. On `hiv_status` the primary's omissions fell from 86 to 60 of 125 while the incumbent's rose from 55 to 72 (Table 1). The missing-extraction difference moved from +0.0022 to −0.0173, lower values favoring the primary; the July primary-minus-incumbent false-extraction difference was +0.0072, with a bootstrap interval excluding zero, indicating more false extraction by the primary. Because the same reference set identified the deficit and assessed the revision, this is a same-set remediation check.

Table 1. Benchmark results on the 398 annotated conversations, all rows scored against the February labels (confidence intervals in supplement Part 7).

Model	Config.	Composite	Critical	`hiv_status`			No record
				Correct	Omis.	Fabr.	
Primary	Feb	0.913	0.879	301/395	86/125	8/270	0
Incumbent	Feb	0.912	0.884	326/395	55/125	10/270	5
Primary	Jul	0.912	0.885	319/395	60/125	16/270	0
Incumbent	Jul	0.913	0.872	309/395	72/125	14/270	0

5.2. *Repeated gating exposed intermittent failures*

At the gating tier each applicable test-programme cell runs twice across up to three deployed programme configurations, and each pair is **stable pass** (both), **flaky** (one), or **stable fail** (neither); these are operational gate labels, not estimated failure probabilities. We drew 420 two-run subsamples from cells run five times. Two runs always matched cells that passed or failed every time, but for a mixed-outcome cell they gave a stable label in roughly half of subsamples and a stable pass in about one in seven (supplement Part 6). Two iterations are therefore an economical screen that can miss intermittent behavior, which is why safety-critical tests use five. Per-run pass rates on the 356 matched cells (712 runs per model) were 92.7% for the primary and 86.1% for the fallback (supplement Table S4). Before a release the full suite is swept up to five times, re-running only tests that have not yet passed. A residual failure, flaky or not, blocks release unless it is a known intermittent failure with no sign of a new regression. The reporter marks more cells flaky than the record check for both models (supplement Table S4), and a date-format validator fix recovered part of the cross-provider gap with no prompt change (supplement Part 3).

5.3. *Error composition changed the failover decision*

Failover decisions must consider error composition as well as aggregate performance. We compared the primary and the deployed cross-provider fallback on 356 matched scenario-programme cells under identical prompts, scaffolding, simulator and scoring. Conversations were regenerated independently in each arm, so the comparison estimates end-to-end behavior rather than transcript-frozen extraction. The primary passed 8.4 points more cells, 90.4% against 82.0% (48 favoring the primary, 18 the fallback). Because a test may contribute cells from more than one programme, cells within a test are not independent, so we report test-clustered inference: a test-cluster bootstrap gave 95% CI +2.3 to +14.7 and a paired test-cluster permutation test gave $p = 0.010$ (the unclustered cell-level exact McNemar gives $p = 0.0003$, which overstates precision).

That aggregate gap understates the difference that matters clinically. Table 2, Panel A, gives the error composition (classes defined in Section 4.2). Over the same 1,342 expected-field comparisons in this matched population the fallback fabricated roughly seven times as often. The ratio remained imprecise because the primary contributed only seven events

across five test clusters (test-cluster bootstrap 95% CI 3.3 to 33.0), while omissions differed much less. A secondary paired cluster permutation test gave p = 0.046 for the difference in error composition. The fallback alone recorded known-value substitutions: in one scenario the client explicitly corrected the assistant, stating that the viral-load result was for hepatitis B and that they did not have HIV, and the fallback still recorded `hiv_status: positive`. The fallback made 20 unsupported `hiv_status` assertions against 1 for the primary, and the primary never substituted a known value in the 356 cells. Of the fallback's 49 fabrications, 41 fall in HIV testing and history (supplement Tables S2 and S5). The gap varied across programmes (+11.9, +8.5 and +4.4 points), and only the largest programme-specific contrast was statistically resolved (p = 0.004, 0.053 and 0.36), so the pooled estimate is not a per-programme guarantee (estimates in the supplement).

The fallback carries every guardrail block and field description, yet shared scaffolding did not make the models interchangeable. The comparison informed fallback choice and monitoring priorities, and supports a future **field-specific failover policy** in which weak safety-critical fields would be set to `unknown` or routed for review until a fallback passes their dedicated tests. That policy is not implemented in production.

5.4. *Scaffolding primarily reduced unsupported extraction*

Guardrail instructions in the prompt and field descriptions in the schema (together, the *scaffolding*) dominate the fixed extraction input, making them natural targets for cost reduction. In a staged ablation over 166 cells across 58 tests, selected by topic rather than outcome, we removed both layers while holding the conversation model, simulator and scenario specifications fixed. Arms without guardrails had every guardrail block removed from the prompt: the 18 core blocks both passes share and the field-specific blocks the EoC pass adds (Section 3.1). Stable-pass rates fell from 87.3% to 63.9% under the bare schema, which retains only field names, types, enums and required flags (test-clustered permutation p = 0.0005), with fabrications rising from 7 to 75 and omissions from 24 to 71 (Table 2, Panel B).

The degradation concentrated in scenarios requiring the model to withhold an unsupported value, while strata whose fields are populated in nearly every conversation changed little (supplement Table S6). The same stripping edit was applied throughout, so the gradient is consistent with differences in what the scenarios require rather than in the manipulation itself. A transcript-frozen replay found the same pattern with conversations held fixed (supplement Part 6), supporting an effect inside the extraction call rather than one mediated by changed conversation generation.

Separating the guidance layers. A second ablation ran all four guidance combinations on a different and smaller population, 119 cells across 42 tests common to every arm, so its contrasts are not comparable in size to the pooled figures above. It separates the compact guardrail blocks from the much larger field-description layer. Removing the guardrails alone cost 16.8 points (p = 0.011). Removing every field description alone had no measurable effect (p = 1.0). Removing both cost 31.1 points (p = 0.0006), 14.3 points more than the two separate effects combined. Field descriptions recovered performance and halved fabrication only once the guardrails were gone, consistent with 37 of the 192 field descriptions embedding

Table 2. Stable-pass rate and field-level error composition (Sections 5.3 and 5.4). Error counts are over field comparisons, so one cell can contribute more than one.

Arm	Cells	Stable-pass	Fabr.	Omis.	Subst.
Panel A: matched failover comparison					
Primary	356	90.4%	7	64	0
Cross-provider fallback	356	82.0%	49	88	14
Panel B: static guidance ablation, primary, 58 tests					
Full scaffolding	166	87.3%	7	24	4
Bare schema	166	63.9%	75	71	2

an explicit prohibition. A single-factor ablation would therefore misclassify the larger layer as expendable.

5.5. *The release gate itself required validation*

Because the preceding analyses rely on the release gate, we evaluated the gate itself, cross-checking the reporter against deterministic comparison of unconditional structured-record expectations. The audit covered 1,369 runs across routine and experimental configurations. Under the repaired comparator, 88 runs passed the reporter while failing the record check: 52 reflected specification, evidence-interface or comparison-rule mismatches, 34 were reporter misses not attributable further, and 2 were reporter assertions unsupported by the evidence supplied, both in ablated configurations. These figures characterize disagreement within the harness, not production error rates.

The audit identified two failure modes: independently maintained prose criteria and structured expectations could drift apart, and some paths did not reliably expose the final profile to the reporter. We synchronized the specifications, supplied the profile directly and added deterministic checks for every record assertion. Adjudication categories, counts and procedure are in supplement Table S7, and the full cross-scorer table in Table S8. The harness engineer performed the adjudication and was not blinded to model arm; the resulting categories should therefore be interpreted as a structured defect analysis rather than an independently validated reference standard.

The corrected gate checks unconditional record assertions deterministically and reserves the reporter for behavior no field encodes, such as whether a response acknowledges a disclosure or provides required safety guidance. The record is checked first: an appropriate response cannot compensate for an incorrect clinical record. An LLM-based verdict reflects not only the evaluator but also the test specification and the evidence supplied to it, so disagreement should be decomposed before it is attributed to the reporter.

Re-running both models under the corrected gate on the same matched 356-cell population, reporter and record checks agree on 91.9% of primary runs and 91.6% of fallback runs, with reporter-pass/record-fail disagreements falling sharply in both (missing-profile rates in supplement Part 6). The model ordering and failover conclusion were unchanged.

6. Discussion and Conclusion

The deployable unit is the complete production configuration, not the model alone. Field-level analysis exposed a `hiv_status` deficit concealed by the aggregate and documented a change in model ordering under a later configuration. A validator change also improved the fallback without a prompt edit. Benchmarking identifies viable candidates; release gating must evaluate the model together with its prompt, schema and validation layer.

The failover comparison shows why a single global ranking is insufficient. The two models differed moderately in stable-pass rate but much more sharply in fabrication and substitution, and those failures do not carry interchangeable consequences in a clinical record. A fallback may remain usable for some fields while safety-critical fields are set to `unknown` or routed for review. That policy is not yet implemented.

No single safeguard accounted for the system's reliability. Guardrails had the larger isolated effect but field descriptions provided backup when guardrails were absent, and selective requiredness, deterministic derivation and field-level recovery protect at different stages. The relevant comparison is therefore between complete configurations, not between base models stripped of the structures they run under.

The release gate had failure modes of its own: some disagreements arose from the reporter, others from drift between prose criteria and structured expectations or from incomplete evidence supplied to it. Assertions encoded in a field should be checked deterministically, and judgment reserved for conversation-dependent behavior the record cannot encode. Together these results support the continuing loop of Figure 2: production failures expand the suite, changes are re-tested, and configuration and instruments are both reviewed.

Future work. Implement the field-specific failover controls these results support, adopt a sequential release rule for high-risk fields, hold out a frozen scenario set, strengthen the reference standard through independent re-annotation, evaluate longitudinal merge correctness and escalation performance, and test transfer across languages and tasks.

7. Limitations

- **Single organization, single annotator; conditional metrics.** The system, schema, metric and data come from one organization; the benchmark covers two South African programmes whereas the deployment spans four, and one annotator per conversation leaves inter-rater agreement unestimated. The post-guidance benchmark reused the same 398 conversations that identified the deficit, so it is a within-sample check rather than validation on unseen conversations, and no scenario set was held out from development, so gating results mix remediation of known failures with generalization. Field-level metrics can be dominated by unknown labels.
- **Scope of the ablation and the model comparison.** The ablation covers mainly HIV treatment and prevention scenarios, plus two safety control strata, where guardrail coverage is densest. The system prompt names only 50 of the 137 declared fields; for the rest the field description may be the only field-specific guidance, so the redundancy claim holds only for the covered fields. The layer-separation runs come from a single earlier batch. All arms regenerated conversations independently, so contrasts include conversational divergence, and

the failover comparison contrasts operational tiers rather than capability-matched models. Two-run classification gives operational labels, not failure probabilities. All transcripts come from one simulator family.

- **The simulator and reporter see the expected record, and the reporter shares a model family with the primary.** The extraction model never receives the expected record. The simulator receives it so the synthetic client makes the intended disclosure, and the reporter so evaluation follows the deployment's clinical rules, which may make scenario pass rates optimistic. Because the conditioning is identical across arms it is more likely to inflate absolute levels than to explain the matched contrasts. The deterministic scorer still scores conversations from that simulator. The reporter, simulator and primary also come from one provider while the fallbacks do not, so reporter-scored cross-provider comparisons carry a possible self-preference bias. Both are why the matched comparison and ablation are scored deterministically.
- **Scope: record correctness, and English-dominant evaluation.** This paper evaluates record correctness and release gating; safety-escalation performance and programme-level service outcomes are reported elsewhere. TB and non-HIV fields are covered by the suite but not separately evaluated, Zimbabwean programme performance is not evaluated on real conversations, and code-switching and non-English disclosure (isiZulu, Shona) remain under-characterized.
- **Longitudinal merge was not evaluated.** Results score the extraction record, before the write into the longitudinal profile. That write is guarded so a known value is not replaced by an `unknown`, but corrections and omissions are not distinguished at merge time.

Artifact availability. The supplement releases the scenario taxonomy and format, five synthetic scenarios, the failure taxonomy, metric definitions, model and sampling configuration and the deterministic comparison rules, supporting an analogous harness: `https://s3.us-west-2.amazonaws.com/fileshare.auderenow.io/public/2026-08-psb2027/clinical-extraction-supplement.pdf`. Conversations, labels, prompts, scorer code and the production harness remain closed for patient-privacy and intellectual-property reasons.

8. Ethics Statement

This analysis used two South African data sources. The first was collected under the *Feasibility and Acceptability Study of an Integrated Multimodal AI Toolkit*, approved by the South African Medical Association Research Ethics Committee (SAMAREC; 280808016/062/2025). The second comprised pseudonymized operational programme data used under Audere's Privacy Policy and applicable data-sharing agreements permitting non-commercial research and development. Before analysis, direct identifiers and potentially identifying free text were removed; retained platform identifiers were pseudonymous and were not linked back to clinical records. No new data were collected for this analysis. Benchmark annotation formed part of routine clinical quality assurance by trained counselors and clinicians. All scenario tests and released artifacts are synthetic and contain no patient data. The Aimee study was funded by the Gates Foundation (INV-067589).

References

1. D. L. Mudzengi, T. Ntshiqa, Y. Hamada *et al.*, Shesha, a WhatsApp chatbot for linking household contacts to tuberculosis treatment or preventive therapy in South Africa: design and development, *JMIR Form. Res.* **9**, e71793 (2025). doi:10.2196/71793.
2. H. Bekele, R. Mendonca, S. Smedinghoff *et al.*, Rising above misinformation: safety-first, quality-assured AI for HIV and sexual and reproductive health self-care, in *AIDS 2026 Abstract Book: 26th International AIDS Conference*, abstract WEPEE587, p. 852 (2026).
3. S. Baptiste, B. Segal, J. Ratevosian *et al.*, From chat to care: translating AI conversations into community-generated program intelligence for HIV services, in *AIDS 2026 Abstract Book: 26th International AIDS Conference*, abstract WEPEE585, pp. 850–851 (2026).
4. N. Mtshali, A. Spyrelis, D. Rech *et al.*, Rethinking entry and access to care: AI-enabled self-care as HIV service infrastructure, in *AIDS 2026 Abstract Book: 26th International AIDS Conference*, abstract WEPEE534, pp. 821–822 (2026).
5. Ö. Uzuner, B. R. South, S. Shen *et al.*, 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text, *J. Am. Med. Inform. Assoc.* **18**(5), 552–556 (2011). doi:10.1136/amiajnl-2011-000203.
6. J.-B. Delbrouck *et al.*, RadGraph-XL: a large-scale expert-annotated dataset for entity and relation extraction from radiology reports, in *Findings of ACL 2024*, pp. 12902–12915. doi:10.18653/v1/2024.findings-acl.765.
7. J. Huang, D. M. Yang, R. Rong *et al.*, A critical assessment of using ChatGPT for extracting structured data from clinical notes, *npj Digit. Med.* **7**, 106 (2024). doi:10.1038/s41746-024-01079-8.
8. Y. Hu, X. Zuo, Y. Zhou *et al.*, Information extraction from clinical notes: are we ready to switch to large language models?, *J. Am. Med. Inform. Assoc.* **33**(3), 553–562 (2026). doi:10.1093/jamia/ocaf213.
9. J. J. Andrew, J. Potier, N. Garcelon *et al.*, Using large language models for temporal relation extraction from pediatric clinical reports, *JAMIA Open* **8**(6), ooaf121 (2025). doi:10.1093/jamiaopen/ooaf121.
10. X. Y. C. Zhang, C. R. Ferreira, F. Rossignol *et al.*, CaseReportBench: an LLM benchmark dataset for dense information extraction in clinical case reports, *Proc. CHIL 2025, PMLR* **287**, 527–542 (2025).
11. T. Schick, J. Dwivedi-Yu, R. Dessì *et al.*, Toolformer: language models can teach themselves to use tools, *Adv. Neural Inf. Process. Syst.* **36**, 68539–68551 (2023).
12. S. G. Patil, T. Zhang, X. Wang *et al.*, Gorilla: large language model connected with massive APIs, *Adv. Neural Inf. Process. Syst.* **37**, 126544–126565 (2024).
13. S. G. Patil, H. Mao, F. Yan *et al.*, The Berkeley Function Calling Leaderboard (BFCL): from tool use to agentic evaluation of large language models, in *Proc. ICML 2025, PMLR* **267**, 48371–48392 (2025).
14. B. T. Willard and R. Louf, Efficient guided generation for large language models (Outlines), arXiv:2307.09702 (2023).
15. Y. Dong, C. Ruan, Y. Cai *et al.*, XGrammar: flexible and efficient structured generation engine for large language models, *Proc. Mach. Learn. Syst.* **7** (2025). arXiv:2411.15100.
16. J. Liu *et al.*, Instructor: structured outputs for LLMs, software library, version 1.17 (2026). `https://github.com/567-labs/instructor` (accessed September 2026).
17. S. Geng, H. Cooper, M. Moskal *et al.*, JSONSchemaBench: a rigorous benchmark of structured outputs for language models, arXiv:2501.10868 (2025).
18. C. Shorten *et al.*, StructuredRAG: JSON response formatting with large language models, arXiv:2408.11061 (2024).

19. N. Ferguson, J. Pennington, N. Beghian *et al.*, ExtractBench: a benchmark and evaluation methodology for complex structured extraction, arXiv:2602.12247 (2026).
20. D. Jin, E. Pan, N. Oufattole *et al.*, What disease does this patient have? A large-scale open domain question answering dataset from medical exams (MedQA), *Appl. Sci.* **11**(14), 6421 (2021). doi:10.3390/app11146421.
21. K. Singhal, S. Azizi, T. Tu *et al.*, Large language models encode clinical knowledge (MultiMedQA / Med-PaLM), *Nature* **620**, 172–180 (2023). doi:10.1038/s41586-023-06291-2.
22. E. J. Gong, C. S. Bang, J. J. Lee *et al.*, Knowledge-practice performance gap in clinical large language models: systematic review of 39 benchmarks, *J. Med. Internet Res.* **27**, e84120 (2025). doi:10.2196/84120.
23. R. K. Arora *et al.* (OpenAI), HealthBench: evaluating large language models towards improved human health, arXiv:2505.08775 (2025).
24. M. Omar, V. Sorin *et al.*, Multi-model assurance analysis showing large language models are highly vulnerable to adversarial hallucination attacks during clinical decision support, *Commun. Med.* **5**, 330 (2025). doi:10.1038/s43856-025-01021-3.
25. B. Atıl, S. Aykent, A. Chittams *et al.*, Non-determinism of "deterministic" LLM system settings in hosted environments, in *Proc. 5th Workshop on Evaluation and Comparison of NLP Systems (Eval4NLP 2025)*, pp. 135–148.
26. D. Wadi and M. Fredette, A Monte-Carlo sampling framework for reliable evaluation of large language models using behavioral analysis, in *Findings of EMNLP 2025*, pp. 9414–9432. doi:10.18653/v1/2025.findings-emnlp.500.
27. M. T. Ribeiro, T. Wu, C. Guestrin *et al.*, Beyond accuracy: behavioral testing of NLP models with CheckList, in *Proc. ACL 2020*, pp. 4902–4912. doi:10.18653/v1/2020.acl-main.442.
28. P. Bhagat, K. N. A. Shastry, P. Panda *et al.*, Evaluating compound AI systems through behaviors, not benchmarks, in *Findings of EMNLP 2025*, pp. 24193–24222. doi:10.18653/v1/2025.findings-emnlp.1314.
29. L. Zheng, W.-L. Chiang, Y. Sheng *et al.*, Judging LLM-as-a-judge with MT-Bench and Chatbot Arena, *Adv. Neural Inf. Process. Syst.* **36** (Datasets and Benchmarks Track), 46595–46623 (2023).
30. Y. Liu, D. Iter, Y. Xu *et al.*, G-Eval: NLG evaluation using GPT-4 with better human alignment, in *Proc. EMNLP 2023*, pp. 2511–2522. doi:10.18653/v1/2023.emnlp-main.153.
31. S. Doddapaneni, M. S. U. R. Khan, S. Verma *et al.*, Finding blind spots in evaluator LLMs with interpretable checklists, in *Proc. EMNLP 2024*, pp. 16279–16309. doi:10.18653/v1/2024.emnlp-main.911.
32. S. Hwang, S. Thomas, H. Williams *et al.*, Leveraging large language models to derive multiple sclerosis progression assessments from clinical notes: a feasibility study, *Pac. Symp. Biocomput.* **31**, 324–337 (2026). doi:10.1142/9789819824755_0023.
33. E. Healey and I. Kohane, LLM-CGM: a benchmark for large language model-enabled querying of continuous glucose monitoring data for conversational diabetes management, *Pac. Symp. Biocomput.* **30**, 82–93 (2025). doi:10.1142/9789819807024_0007.
34. P. Budzianowski, T.-H. Wen, B.-H. Tseng *et al.*, MultiWOZ: a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling, in *Proc. EMNLP 2018*, pp. 5016–5026. doi:10.18653/v1/D18-1547.
35. T. Kim, H. Yoon, Y. Lee *et al.*, Mismatch between multi-turn dialogue and its evaluation metric in dialogue state tracking, in *Proc. ACL 2022 (Short Papers)*, pp. 297–309. doi:10.18653/v1/2022.acl-short.33.
36. A. Spyrelis, S. Frade, S. Cooper *et al.*, Aimee, an AI-powered HIV prevention and sexual and reproductive health intervention for adolescent girls and young women in South Africa: design, pilot study methods, and feasibility [version 1; preprint, awaiting peer review], *VeriXiv* **3**, 287 (2026). doi:10.12688/verixiv.3679.1.