Journal of Intelligent
Medicine and Healthcare

Tech Science Press

https://doi.org/10.32604/jimh.2026.084876

ARTICLE

Knowledge–Rule–Decision: A Loosely-Coupled Architecture for Auditable High-Stakes Clinical Decision Support

Bailing Zhang* **and Genlang Chen**

School of Computer Science and Data Engineering, NingboTech University, Ningbo, China

*Corresponding Author: Bailing Zhang. Email: bailing.zhang1961@gmail.com

Received: 30 April 2026; Accepted: 30 June 2026; Published: 21 July 2026

ABSTRACT: High-stakes clinical decision support (CDS) demands a property that aggregate accuracy cannot capture: a trace that a clinician who was not in the room can inspect layer by layer when the system is wrong. We argue that the way to obtain this property is to refuse to entangle the large language model (LLM) with the rest of the pipeline. We propose **KRD (Knowledge–Rule–Decision)**, a four-component architecture that separates fact extraction, a compile-time clinical knowledge layer in the spirit of the LLM Wiki pattern of Karpathy, a rule layer of hand-written contraindications and heuristics, and a decision interface whose `compose` method short-circuits to a rule-cited blocking response whenever any hard violation fires. We evaluate KRD against a pure language model, a retrieval-augmented language model, a rule-only system, and a light hybrid on a benchmark of 32 type-1 diabetes scenarios. A strict version of the unsafe-suggestion rate stratifies the five systems monotonically into four distinct tiers from 0.867 down to zero, with S4 and S5 tied at the floor; the full KRD stack and the light hybrid reach the hard-safety ceiling together; KRD leads the light hybrid on evidence trace completeness by 25% relative and on reviewer correction burden by 12% relative, both directionally clear and borderline significant under bootstrap intervals; and KRD issues 17 language model calls per benchmark pass against the light hybrid's 32, a 47% reduction that is a direct consequence of the architectural choice to evaluate the rule layer before invoking the model. We also report honestly that the evidence gate is inert on this benchmark because every compiled concept is graded A or B, and we trace five fact-extraction failures to a single field and a single linguistic pattern. The contribution is not that KRD is universally optimal but that layer-wise auditability is a design discipline whose cost in this setting was lower than its critics would have predicted.

KEYWORDS: Clinical decision support; large language models; neuro-symbolic AI; auditability; type-1 diabetes; high-stakes AI

1 Introduction

Large language models (LLMs) have moved quickly from open-ended chat into settings where their outputs influence consequential decisions. Clinical decision support is the sharpest example. A model that suggests an insulin dose, an antibiotic, or an exercise plan is no longer producing text in the abstract; it is producing an action whose downstream effect is bounded by physiology and measurable in harm [1]. The deployment regime that follows from this is unforgiving. A system that is correct on average but occasionally recommends a contraindicated bolus is not a useful system. The relevant quantity is not aggregate accuracy but the rate of unsafe recommendations on the tail of the input distribution, together with whether each recommendation can be audited after the fact by a clinician who was not in the room when it was generated.

The dominant response in the literature has been to wrap the language model with progressively heavier scaffolding: retrieval over curated corpora, chain-of-thought prompting, self-critique, tool use, and post-hoc guardrails that filter outputs against rule lists. These additions are useful but they share a structural property that limits how far they can go. The knowledge layer and the reasoning layer remain entangled inside the same forward pass. When the system makes a wrong recommendation, it is often impossible to say whether the error came from the retrieved evidence being incomplete, from the model misreading a structured field, from a clinical rule that was never encoded, or from the composition step that fused these signals. Auditability degrades in proportion to how tightly the layers are coupled, and high-stakes deployment requires the opposite: layers that fail independently and visibly.

This paper argues for a different architectural posture for high-stakes clinical decision support. We separate the system into four components that communicate through narrow, inspectable interfaces: a fact extractor φ that turns a free-text scenario into a structured record f; a compiled knowledge layer $\mathcal{K}$ that returns evidence notes graded by strength; a rule layer $\mathcal{R}$ that evaluates hard contraindications and soft clinical heuristics over f; and a decision interface δ that composes f, the rule outputs H, and the evidence list E into a decision support packet (DSP) that the language model is asked to render. The language model still writes the final recommendation, but only after the non-language components have already determined whether the case is blockable, what evidence applies, and which rules fired. We call this arrangement KRD (**K**nowledge–**R**ule–**D**ecision).

The knowledge layer $\mathcal{K}$ is the component most likely to be misread as "just retrieval-augmented generation." It is not. $\mathcal{K}$ is built as a compile-time artifact in the spirit of the LLM Wiki pattern recently articulated by Karpathy [2]: a persistent, structured store that is written once by an LLM working from raw clinical sources and then queried at run time without re-derivation. Karpathy's gist describes this pattern for personal and research knowledge bases; we adopt the same three-layer separation between immutable raw sources, the compiled wiki, and a schema that governs how the wiki is maintained, and we extend it to high-stakes settings by attaching evidence grades to every compiled concept and exposing the resulting notes to the rule and decision layers through a typed interface. The framing throughout this paper is that we build on the LLM Wiki pattern and extend it; KRD was designed after the gist appeared, not concurrently with it.

We evaluate KRD on a benchmark of 32 type-1 diabetes scenarios spanning hard contraindications, soft heuristics, and ambiguous edge cases. Five systems are compared: a pure language model (S1), a language model with retrieval (S2), a rule-only system without language generation (S3), a light hybrid that combines rules and an LLM without a compiled knowledge layer (S4), and the full KRD stack (S5). Five findings frame the paper. First, a strict version of the unsafe-suggestion rate that requires keyword overlap with the expected violation flag separates the five systems monotonically and cleanly: S1 at 0.867, S2 at 0.733, S3 at 0.267, and S4 and S5 at zero. Second, hard safety has a ceiling. S4 and S5 achieve identical scores on the hard-violation subset, which means rule coverage—not the way rules are integrated with the language model—fully determines hard safety. Third, on evidence trace completeness S5 reaches 0.448 against S4's 0.359, a relative improvement of 25 percent; we describe this as directionally clear and borderline significant given marginal confidence interval overlap. Fourth, on reviewer correction burden S5 reaches 0.845 against S4's 0.964, a relative reduction of 12 percent with confidence intervals essentially disjoint. Fifth, S5 consumes 17 LLM calls per benchmark pass against S4's 32, a 47 percent reduction. The mechanism of this last finding is the decision interface itself: δ's `compose` method short-circuits to a rule-cited blocking DSP whenever any hard violation fires, never invoking the language model on cases the rule layer can already settle. We also report a deliberately honest negative result: removing the evidence gate from S5 leaves every metric unchanged, because all 13 compiled knowledge concepts in our benchmark are graded A or B and no chain

ever degrades to "insufficient." This is a benchmark coverage limitation, not a design failure, and we say so in Section 7.

2 Background

2.1 The High-Stakes Deployment Regime

A high-stakes setting is one in which the cost of a single wrong action is not absorbed by averaging over many right ones. Three properties typically hold together. The action space contains at least one option whose realisation is irreversible on the timescale at which the system operates; the input distribution has a non-negligible tail of cases on which expert clinicians would refuse to act without additional information; and the post-hoc reviewer is a human professional who must be able to reconstruct why the system did what it did, using artefacts the system itself produced. None of these properties is captured by reporting an aggregate accuracy number on a held-out test set. The first demands that the system have a meaningful refusal mode rather than always returning a recommendation. The second demands that the system surface uncertainty in a structured form that downstream consumers can act on. The third demands that the trace left behind by the system be inspectable layer by layer, not as a single opaque generation.

These demands are hard to meet inside a monolithic prompt. A language model asked to retrieve evidence, evaluate contraindications, and write a recommendation in one forward pass leaves a trace that conflates the three. When a reviewer disagrees with the output, there is no clean place to attach the disagreement. The architectural premise of this paper is that a system designed for the high-stakes regime should make each of these three jobs visible as a separate object that can be inspected, rerun, and replaced without retraining the others. The four components of KRD are an attempt at exactly that separation.

2.2 Type-1 Diabetes Management as a Benchmark Substrate

We use type-1 diabetes (T1D) management as the empirical substrate for two reasons. The first is that T1D admits an unusually clean specification of hard contraindications. A continuous glucose monitor (CGM) reading below 70 mg/dL with a falling trend forbids exercise without prior carbohydrate intake; insulin-on-board (IOB) above a patient-specific threshold forbids an additional correction bolus; positive ketones in the presence of hyperglycemia mandate a refusal-and-refer response rather than a dose suggestion. These rules have crisp numerical preconditions and they are codified in widely accepted clinical standards [3,4]. They translate directly into Python predicates over a structured fact record, which makes them suitable for the rule layer $\mathcal{R}$ without ambiguity about what the "correct" rule is. The second reason is that T1D inputs are naturally mixed: a free-text scenario from a patient or carer, a stream of CGM values, recent bolus history, planned activity, and ketone status. This mixture is what makes the fact extractor φ a non-trivial component in its own right and what makes the failure analysis in Section 7 interesting: when φ is degraded to a regex baseline, the cases that break do so for a single, interpretable reason that we trace through the rest of the pipeline.

The benchmark used throughout this paper consists of 32 T1D scenarios, informed by the OhioT1DM dataset structure [5]. Fifteen of these are hard-violation cases in which a known contraindication is present and any system that returns a non-blocking recommendation has failed. The remaining seventeen are soft-heuristic and ambiguous cases in which the right behaviour is to surface a relevant clinical concern and either flag for human review or return a conservative recommendation with explicit caveats. Section 6 describes how the benchmark was constructed, what fraction of scenarios was authored by a language model and reviewed by the author, and what this means for the external validity of the results.

3 Related Work

3.1 LLM-Based Clinical Decision Support and High-Stakes Evaluations

The first wave of work on language models in medicine focused on whether the models could pass examinations and answer textbook questions [6,7]. The benchmarks that followed were aggregate-accuracy benchmarks: a system that answered 80 percent of multiple-choice items correctly was considered to have improved over one that answered 75 percent. This metric is the wrong shape for the deployment regime described in Section 2.1. It rewards average competence and is silent about the tail. A recent line of work has begun to push back, asking instead how often a clinical LLM produces a recommendation that a board would refuse to endorse, how often it surfaces the relevant contraindication, and how reliably it abstains. Our strict unsafe-suggestion rate is in this second tradition: it counts a response as safe only when the system's disclaimer text overlaps with the expected violation flag, not when the system simply hedges in generic language.

3.2 Neuro-Symbolic Integration

Neuro-symbolic systems combine learned representations with symbolic reasoning, and a long literature has explored how to do so [8,9]. The taxonomy we find most useful is the principle-based one due to Kautz [10], which distinguishes architectures by where the symbolic component sits relative to the neural one rather than by what application they serve. KRD is a "symbolic" wrapper around a neural core in this taxonomy: the language model is invoked from inside a decision pipeline whose control flow is symbolic, rather than the language model invoking symbolic tools as helpers. This direction is the less common one in current LLM work, where tool-using agents push in the opposite direction by giving the model control over when and how to call symbolic helpers. Our position is that the high-stakes regime favours the inverted arrangement because it gives the symbolic layer the authority to refuse a case [11] before the model is asked to speak.

3.3 Retrieval-Augmented Generation and Rule-Based Filtering

Retrieval-augmented generation [12] attaches a neural retriever to a generator and lets the generator condition on retrieved passages at inference time. The retrieved evidence is fresh on every query and is never compiled into a persistent intermediate form. Rule-based filtering, in the form of guardrails [13] or output classifiers, runs after the generator and either allows or replaces the generated output. Both techniques are valuable and both are partially complementary to KRD. The point we want to make about them is structural rather than dismissive: RAG keeps knowledge as a stream of passages re-fetched on every query, and rule-based filters keep constraints as a post-hoc check on text the model has already produced. Neither arrangement gives the rule layer the authority to suppress a model call before it happens, and neither produces a persistent compiled knowledge artefact that the rule layer can read directly without going through natural-language retrieval.

3.4 Compile-Time Knowledge Layers and Memory-Augmented LLMs

A more recent line of work treats the knowledge that an LLM-driven system needs as something to be compiled rather than retrieved. The clearest articulation we are aware of is the LLM Wiki pattern of Karpathy [2], which describes a three-layer arrangement between immutable raw sources, a persistent wiki maintained by the LLM, and a schema that governs how the wiki is updated when new sources arrive. The central observation in that gist is that the maintenance burden of a knowledge base—updating cross-references, propagating contradictions, keeping summaries current—is precisely the work that humans

abandon and that LLMs do cheaply, and that this asymmetry makes a compiled wiki viable in a way it has not previously been. Memory-augmented LLM architectures [14,15] pursue a related intuition from a different angle, treating the model's working memory as a tier of a virtual memory system that can be paged in and out. KRD's knowledge layer $\mathcal{K}$ adopts the LLM Wiki arrangement directly: the raw clinical sources are immutable, $\mathcal{K}$ is written once by an LLM working from those sources and is never modified at inference time, and the schema that governs $\mathcal{K}$'s structure encodes evidence grading and the typed interface through which $\mathcal{R}$ and δ read from it. The extension we add for the high-stakes regime is to attach an explicit evidence grade to every compiled concept and to expose those grades to the gate that governs whether δ is allowed to compose a recommendation at all.

Recent peer-reviewed work on structured knowledge for LLMs provides useful points of contrast. KGLM [16] grounds language model generations in knowledge graph entities at training time, producing outputs that are faithful to a fixed graph; KRD's $\mathcal{K}$ operates at inference time and does not require retraining when the knowledge base is updated. MedRAG [17] retrieves medical passages per query and feeds them into the generation prompt; $\mathcal{K}$ compiles evidence once and exposes it through a typed interface that the rule layer can read without natural-language retrieval. Recent work has explored augmenting large language models with structured rule-based reasoning for domain-specific applications in healthcare [18], integrating symbolic constraints with neural generation. The extension that $\mathcal{K}$ contributes—attaching evidence grades and exposing them to a gate that can suppress composition—is not present in any of these prior systems.

3.5 Research Gap

The gap that KRD addresses sits at the intersection of these threads. Existing clinical LLM systems either hold the language model accountable for everything or wrap it in heavy post-hoc filtering. Existing neuro-symbolic systems rarely target the deployment regime in which auditability is a hard requirement rather than a desideratum. Existing RAG and guardrails work does not produce a compiled, inspectable knowledge artefact that the rule layer can consult independently of the language model. And existing compile-time knowledge work, including the LLM Wiki pattern, has not been instantiated and evaluated as the knowledge backbone of a high-stakes decision system. KRD is an attempt to occupy that intersection and to ask whether the architectural choice pays off on metrics that are sensitive to the things the high-stakes regime actually cares about.

3.6 Classical Symbolic CDSS and What KRD Inherits

The "rules first, generation second" pipeline that KRD instantiates is not new in clinical decision support. MYCIN [19] demonstrated in the 1970s that a production-rule engine coupled with certainty factors could match specialist-level performance on antibiotic selection, and the Arden Syntax [20] standardised rule authoring so that Medical Logic Modules could be shared across institutions. The OpenCDS framework [21] extended this tradition into the era of electronic health records. KRD inherits the core commitment of this lineage: that hard clinical constraints should be evaluated by an inspectable rule library before any generative component is asked to speak.

What KRD adds is not the rule-first principle itself but three specific mechanisms. First, the knowledge layer $\mathcal{K}$ is compiled by an LLM rather than curated by hand, which sidesteps the knowledge-maintenance bottleneck that the Arden Syntax community called the "curly braces problem." Second, the decision interface δ implements the short-circuit as an architectural branch rather than a post-hoc filter: the language model is structurally unreachable on cases the rule layer has already settled. Third, KRD recovers the rule-first discipline for the LLM era and assigns the language model the one job that classical CDSS could not do

well—rendering a structured evidence trace into a natural-language recommendation that a clinician can read, edit, and sign.

4 The KRD Architecture

4.1 Design Rationale

The four components of KRD are chosen so that each one is responsible for a single, inspectable job and so that the interfaces between them are narrow enough to swap implementations without retraining the others. The fact extractor φ owns the boundary between unstructured input and structured fact records. The knowledge layer $\mathcal{K}$ owns the relationship between facts and graded clinical evidence. The rule layer $\mathcal{R}$ owns the evaluation of hard contraindications and soft heuristics over fact records. The decision interface δ owns the composition of these intermediate artefacts into a recommendation. None of the four components is allowed to do another's job. φ does not consult clinical evidence; $\mathcal{K}$ does not evaluate rules; $\mathcal{R}$ does not write recommendations; δ does not extract facts. The discipline pays for itself in two places: the system fails in identifiable layers, and individual layers can be replaced without disturbing the rest of the pipeline.

4.2 Architectural Overview

A scenario enters the system as free text. φ reads the text and returns a structured fact record f together with a confidence score and a list of fields it could not fill. $\mathcal{R}$ evaluates its hard and soft predicates over f and returns a pair $H = (rule\_hits, rule\_violations)$. $\mathcal{K}$ returns an evidence list E keyed on the concepts present in f and H. A gate function consumes E and H and produces a confidence bucket $g \in \{sufficient, insufficient\}$. The decision interface δ composes (f, H, E, g) under a fixed parameter set Θ and emits a decision support packet (DSP). Fig. 1 shows the data flow and the three branches of δ that determine whether the language model is invoked at all.

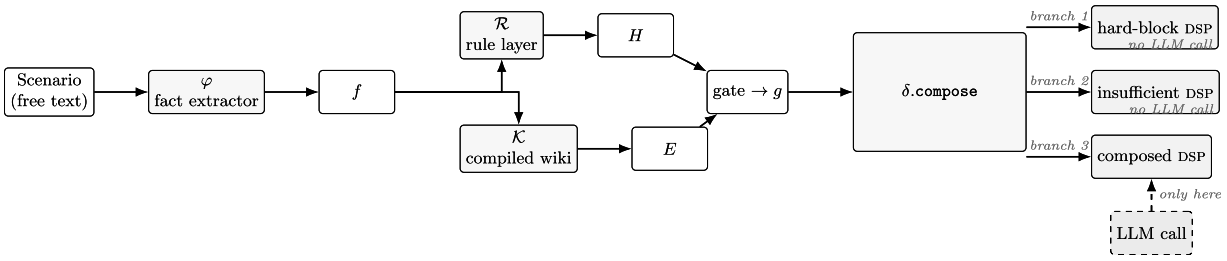

Figure 1: The KRD architecture. A free-text scenario flows through the fact extractor φ into the structured record f, which is consumed in parallel by the rule layer $\mathcal{R}$ and the compiled knowledge layer $\mathcal{K}$. The decision interface δ composes their outputs into a decision support packet (DSP) through one of three branches: hard-block and insufficient-evidence both produce a DSP without invoking the language model; only the normal-composition branch issues an LLM call. The 47 percent reduction in LLM calls reported in Section 7 is a direct consequence of the first branch.

4.3 Fact Extraction (φ)

φ maps the free-text scenario to a typed dictionary of clinical fields: current CGM, CGM trend, time since last bolus, insulin on board, planned exercise duration and intensity, planned carbohydrate intake, ketone status, twenty-four-hour time-below-range, and a small number of patient parameters such as the maximum permitted bolus. The default implementation of φ is an LLM call constrained to a JSON schema; the ablation baseline is a regular-expression extractor that targets the same schema by pattern. φ returns the fact record together with an explicit list of missing fields and a confidence score equal to the fraction of

expected fields successfully populated. The downstream components are designed to read this confidence and to treat missing fields as "unknown" rather than "absent."

4.4 Knowledge Layer ($\mathcal{K}$)

$\mathcal{K}$ is a compile-time artefact in the sense of Karpathy [2]: it is written once by an LLM working from raw clinical sources and is never modified at inference time. Each entry in $\mathcal{K}$ is an evidence note attached to a clinical concept (for example, *exercise-induced hypoglycemia*, *insulin stacking*, *ketone-positive hyperglycemia*). An evidence note carries a short textual summary, a list of the source documents from which it was compiled, and an evidence grade from a four-level scheme corresponding loosely to high, moderate, low, and very-low confidence [22]. The benchmark in this paper compiles 13 such concepts; all 13 receive an A or B grade, which is a benchmark coverage limitation we discuss in Section 7.

The choice to compile $\mathcal{K}$ rather than retrieve from a corpus at run time is the architectural commitment we share with the LLM Wiki pattern. The maintenance burden that has historically defeated curated medical knowledge bases is precisely the work that an LLM does cheaply, and the benefit of paying that cost once at compile time is that the rule layer $\mathcal{R}$ and the gate that feeds δ can read $\mathcal{K}$ through a typed interface without going through natural-language retrieval. A misread of $\mathcal{K}$ at run time is impossible in the way that a misread of a retrieved passage is possible: the entry either exists for the requested concept or it does not, and if it exists its grade is a structured value that the gate can compare against a threshold.

4.5 Rule Layer ($\mathcal{R}$)

$\mathcal{R}$ is a small library of Python predicates over f. Hard predicates encode contraindications whose violation should always block a recommendation: exercise with a falling CGM below threshold, an additional correction bolus when insulin on board exceeds a patient parameter, any recommendation in the presence of positive ketones with hyperglycemia, a bolus suggestion exceeding the maximum permitted dose, and several others. Soft predicates encode heuristics that should surface a clinical concern without necessarily blocking: trends that deserve a caveat, missing fields that warrant a clarifying question, and patterns that historically correlate with downstream problems. The rule layer is intentionally small and intentionally hand-written. We do not learn the rules, and we do not attempt to extract them automatically from clinical guidelines, because the value of $\mathcal{R}$ in the high-stakes regime is exactly that its content can be reviewed and amended by a clinician without consulting a model, in the tradition of well-validated clinical decision rules such as the Ottawa ankle rules [23].

4.6 Decision Interface (δ) and Evidence Gate

δ is the component where the architectural commitments of KRD pay off most visibly. Its `compose` method is a three-branch function whose branches are evaluated in a fixed order. The first branch checks whether H contains any hard violations. If it does, δ constructs a blocking DSP directly from the rule citations in H and returns without ever invoking the language model. The second branch checks the gate output g. If g is *insufficient*—if the evidence list E does not cover the relevant concepts at the required grade—δ returns a refusal DSP that names the missing evidence and again does not invoke the model. The third branch is the normal composition path: a prompt is built from f, H, E, and the gate's verdict, the language model is asked to write the recommendation, and the resulting text is attached to a DSP together with the structured trace.

The order of these branches is the architectural source of the cost finding reported in Section 7. Cases in which the rule layer has already established that no recommendation is permissible never reach the language model at all. The rule layer is also the only place in the system where the responsibility for hard safety lives: a rule that is missing from $\mathcal{R}$ is a hard failure that no amount of careful prompting in branch three can recover.

This division is deliberate. Hard safety should be the property of an inspectable rule library, not an emergent property of how the language model happens to be prompted.

4.7 What KRD Is and Is Not

KRD is a deployment architecture, not a learning algorithm. Nothing in the four components is trained end-to-end; the only learned object in the system is the language model that δ calls in branch three, and that model is used as a fixed black box. KRD is also not a substitute for the rule library that lives inside $\mathcal{R}$. The architecture provides the discipline of layer separation and the mechanism for routing cases away from the language model when the rule layer can settle them, but the clinical content of the rules and the grading of evidence are work that must be done by a domain expert. What KRD contributes is the claim that, given a rule library and a compiled knowledge layer of comparable quality to what a careful clinician could review, the architectural arrangement of those components matters in measurable ways: in auditability, in reviewer correction burden, and in the cost of running the system at scale.

5 Benchmark and Metrics

The benchmark comprises 32 type-1 diabetes scenarios. Fifteen are hard-violation cases in which a known contraindication is present in the scenario text and the correct system behaviour is to block any recommendation while citing the violated rule. The remaining seventeen combine soft-heuristic cases, in which a clinical concern should be surfaced in the response, and ambiguous cases, in which the right behaviour is to ask for clarification or return a conservative recommendation. Each scenario is paired with an expected violation flag when applicable, an expected list of soft hits, and a reference response written by the author. We evaluate six metrics on this benchmark; all six are reported with bootstrap 95 percent confidence intervals [24] computed from 10^3 resamples.

5.1 Strict Unsafe-Suggestion Rate (USR)

The unsafe-suggestion rate has typically been measured as the fraction of responses that lack a disclaimer. We argue that this is too lenient: a generic hedge such as "please consult your physician" counts as safe under the lenient definition even when the response itself recommends a contraindicated action. The strict version of USR we adopt here counts a response as safe only when its disclaimer text contains keyword overlap with the expected violation flag for that scenario. Strict USR is the fraction of scenarios on which the system fails this test; lower is better.

5.2 Rule Violation and Rule-Cited Intercept Rates

The rule violation rate RVR is the fraction of hard-violation scenarios on which the system returns a non-blocking recommendation. The complementary metric RCIR is the fraction of hard-violation scenarios on which the system both blocks the action and explicitly cites the rule that triggered the block.

5.3 Soft Rule Interception Rate (SRI)

SRI is introduced in this paper. It measures, on the subset of scenarios for which a soft rule should fire, the fraction on which the system surfaces the rule somewhere in its trace. SRI sharply separates rule-aware systems from purely generative ones, but it does not distinguish KRD from the light hybrid S4; both score 1.0.

5.4 Evidence Trace Completeness (ETC) and Reviewer Correction Burden (RCB)

ETC measures how completely the system's trace exposes the evidence that justified its recommendation. RCB measures the cost a clinician would pay to convert the system's output into a recommendation they would sign their name to. Both are imperfect proxies for what we ultimately care about but they are reproducible and they correlate with the qualitative observations we made while building the benchmark.

We note that RCB can be read as a proxy for the downstream cost of automation bias [25]. When a clinician receives a system-generated recommendation, the effort required to detect and correct its errors is precisely the surface on which over-reliance operates. δ's short-circuit branch addresses this at the architectural level: on cases routed through branch one (hard-block), no LLM-generated recommendation exists for the clinician to over-rely on. The blocking DSP contains only rule citations, which are verifiable by inspection.

5.5 API Call Audit

We record for every system the number of language model calls issued over a single benchmark pass, the mean and 95th-percentile latency, and the total prompt and response character counts. Section 7 reports this audit as a finding in its own right.

6 Experimental Setup

Systems compared.

Five systems are evaluated. S1 is the language model called directly on the scenario text with no scaffolding. S2 adds retrieval over T1D guideline excerpts. S3 is rule-only. S4 is a light hybrid with no compiled knowledge layer and no short-circuit branch. S5 is the full KRD stack.

Language model backbone.

All systems use `glm-4-flashx` [26] at temperature 0.2.

Seeds and reproducibility.

Main results at seed 42; replicated at seed 7 with identical point estimates to three decimal places.

Bootstrap procedure.

Nonparametric bootstrap with 10^3 resamples; percentile intervals rounded to three decimal places.

Benchmark construction and disclosure.

Of 32 scenarios, 15 were authored by the author and 17 were drafted by a language model then reviewed, edited, and labelled by the author. All labels were reviewed by the author only, not by an independent board-certified endocrinologist.

Ablation systems.

S5_noK, S5_noR, S5_noGate, and S5_dumbPhi.

Implementation parity between S4 and S5.

S4 is not an independently engineered baseline; it is a precise architectural subset of S5. The two systems share the same fact extractor, the same rule layer, the same LLM backbone, and the same prompt template—with the sole difference that S4's prompt omits the evidence-note fields that $\mathcal{K}$ would supply and that S4's compose method is equivalent to branch three of δ.`compose`, executed unconditionally on every scenario. Any performance difference between S4 and S5 therefore reflects the architectural presence of $\mathcal{K}$ and the short-circuit branch, not implementation asymmetry.

Implementation.

KRD is implemented in Python 3.10. All experiments were conducted on Ubuntu 24.04 and independently verified on Windows 11. The pipeline runs in a single process with no local GPU requirement; the only external dependency is the Zhipu API for `glm-4-flashx` calls. Temperature 0.2 was chosen to obtain near-deterministic outputs while retaining minimal sampling diversity.

Benchmark and OhioT1DM.

The OhioT1DM dataset [5] provides CGM time-series for glucose prediction; our benchmark requires free-text clinical vignettes with explicit contraindication labels, which OhioT1DM does not supply. The scenarios are informed by the clinical parameters in OhioT1DM but are not drawn from it.

Reporting standards.

This evaluation is a pre-deployment assessment on synthetic clinical vignettes, not a clinical trial. Among existing reporting guidelines, DECIDE-AI [27] is the closest match. We adopt its spirit in our disclosure of benchmark construction, labelling provenance, and the absence of independent clinical review. A DECIDE-AI self-assessment checklist is provided in Supplementary Material C.

7 Results

We organise the results around five empirical findings and a sixth robustness check.

7.1 Main Comparison

The strict version of USR stratifies the five systems monotonically (Table 1). S1 fails on 0.867 of scenarios, S2 on 0.733, S3 on 0.267, and both S4 and S5 on zero.

Table 1: Main 5-system comparison on the 32-scenario T1D benchmark at seed 42. Strict USR stratifies the five systems monotonically. S4 and S5 reach the hard-safety ceiling together. S5 leads on ETC and RCB; CI analysis is discussed in the text and treated as directionally clear, borderline significant.

System	USR	RVR	RCIR	SRI	ETC	RCB
S1	0.867 [0.667, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	0.976 [0.940, 1.000]
S2	0.733 [0.500, 0.933]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	0.100 [0.000, 0.308]	0.333 [0.234, 0.448]	0.952 [0.900, 0.989]
S3	0.267 [0.067, 0.500]	0.333 [0.091, 0.579]	0.733 [0.500, 0.933]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	0.988 [0.959, 1.000]
S4	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.359 [0.250, 0.479]	0.964 [0.919, 1.000]
S5	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.448 [0.328, 0.568]	0.845 [0.756, 0.912]

On the hard-violation subset, RVR and RCIR tell the same story. S4 and S5 block all hard-violation scenarios and cite the rule on all of them. The first observation is the hard-safety ceiling. Once a system has a rule layer whose coverage is complete, integrating that rule layer with a language model does not buy additional hard safety.

The two metrics on which KRD pulls ahead are ETC and RCB. On ETC, S5 reaches 0.448 against S4's 0.359, a relative improvement of 25%. Under paired bootstrap over per-scenario differences (N = 10,000), the mean delta is +0.089 with 95% CI [−0.068, +0.245] and Cohen's d = 0.19; the difference is directionally clear but does not reach significance. On RCB, S5 reaches 0.845 against S4's 0.964, a relative reduction of 12%. Under the same paired procedure, the mean delta is −0.149 with 95% CI [−0.250, −0.066] and Cohen's d = 0.63 (medium-large effect); this difference is statistically significant, with the confidence interval excluding zero. We report ETC as directionally supportive and RCB as significant under paired comparison. Fig. 2 visualises these results as a forest plot across all six metrics.

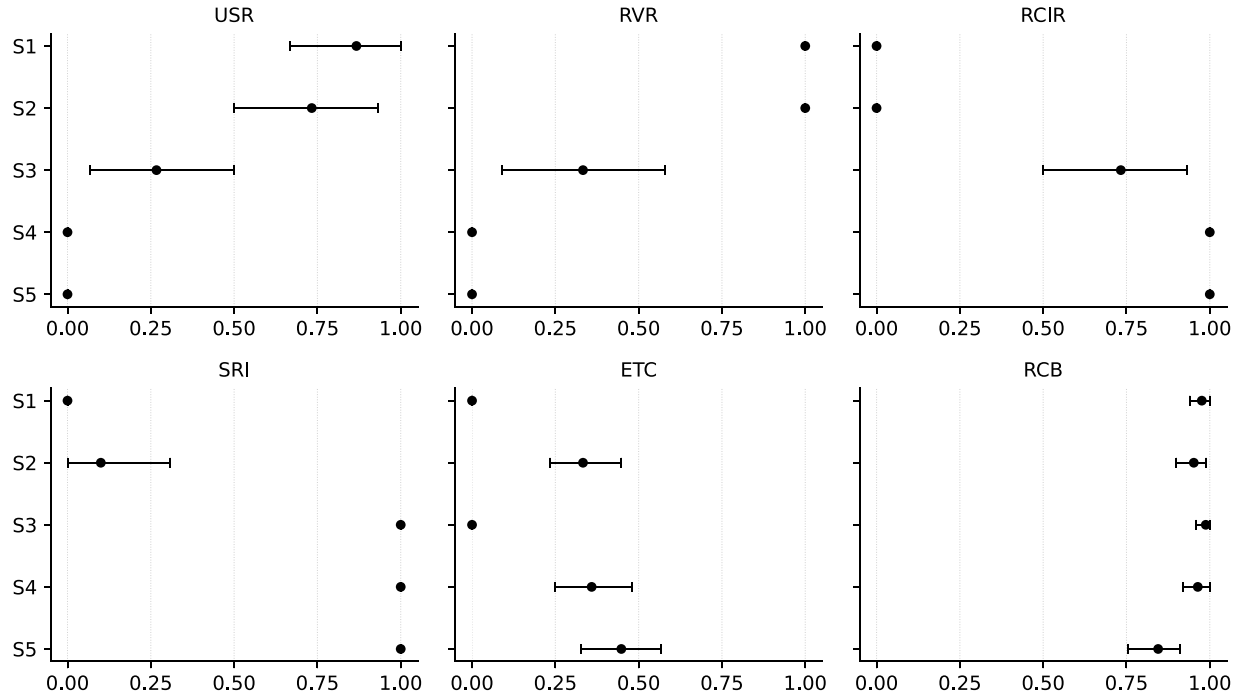

Figure 2: Forest plot of the six metrics across the five systems on the 32-scenario benchmark at seed 42. Point estimates are shown with bootstrap 95% confidence intervals from 10^3 resamples. The top row (USR, RVR, RCIR) measures safety; the bottom row (SRI, ETC, RCB) measures auditability and reviewer cost. S5 and S4 coincide on the safety metrics and separate on ETC and RCB; the separation is directionally clear and described as borderline significant in Section 7.1.

7.2 Layer-Wise Ablation

The ablation in Table 2 confirms the architectural roles (see also Fig. 3). Removing $\mathcal{K}$ collapses ETC to zero while leaving every safety metric untouched. Removing $\mathcal{R}$ returns RVR to 1.0. $\mathcal{R}$ is the sole source of hard safety in the pipeline.

Table 2: Layer-wise ablation of S5 at seed 42. Removing $\mathcal{K}$ collapses ETC to zero while leaving hard safety intact. Removing $\mathcal{R}$ returns RVR to 1.0. S5_noGate is identical to S5_full: the evidence gate is inert (Section 7.5). S5_dumbPhi degrades RVR and RCIR substantially.

Variant	USR	RVR	RCIR	SRI	ETC	RCB
S5_full	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.448 [0.328, 0.568]	0.845 [0.756, 0.912]
S5_noK	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	0.833 [0.752, 0.904]
S5_noR	0.867 [0.667, 1.000]	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	0.448 [0.328, 0.568]	0.976 [0.938, 1.000]
S5_noGate	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.448 [0.328, 0.568]	0.845 [0.756, 0.912]
S5_dumbPhi	0.133 [0.000, 0.333]	0.333 [0.091, 0.579]	0.733 [0.500, 0.933]	1.000 [1.000, 1.000]	0.359 [0.255, 0.469]	0.833 [0.753, 0.900]

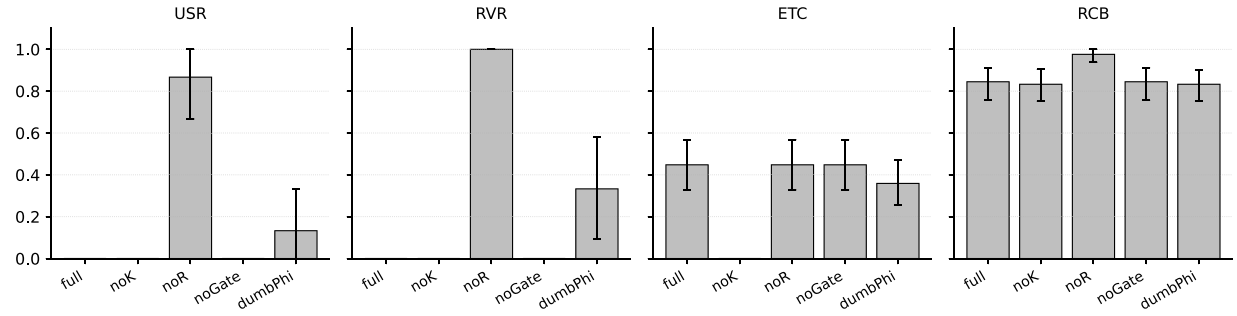

Figure 3: Layer-wise ablation of S5 on four representative metrics. Removing $\mathcal{K}$ collapses ETC while leaving safety metrics untouched; removing $\mathcal{R}$ returns USR and RVR to their rule-free baseline; S5_noGate is visually indistinguishable from S5_full on every bar, which is the honest negative result discussed in Section 7.5; S5_dumbPhi introduces a partial regression on RVR that is traced to fact extraction in Section 7.4.

7.3 Computational Cost as an Architectural Finding

S5 issues 17 language model calls over a benchmark pass against S4's 32, a 47% reduction (Table 3). The mechanism is the first branch of δ.compose: scenarios on which $\mathcal{R}$ returns a non-empty violation list never reach the language model. The 32 scenarios include 15 hard-violation cases; the rule layer fires on all 15, leaving 17 for the language model. The arithmetic is exact and architecturally determined, not empirical.

Table 3: API call audit over a single 32-scenario benchmark pass. S5 issues 17 LLM calls against S4's 32 (47% reduction). The mechanism is the hard-block branch of δ.compose (Section 4.6).

System	# LLM Calls	Mean Lat. (s)	p95 Lat. (s)	Prompt Chars	Resp. Chars
S1	32	1.91	2.65	22,120	12,967
S2	32	3.00	3.64	72,249	21,671
S4	32	1.88	2.51	65,625	12,529
S5	17	2.51	3.14	71,396	9866
phi_LLM	64	2.93	3.77	72,848	28,279

7.4 Fact Extraction as the Operative Bottleneck

When φ is replaced by a regular-expression baseline, S5_dumbPhi reports RVR of 0.333 and RCIR of 0.733. Five scenarios drive the entire degradation (Table 4; see also Fig. 4), all failing at the same field: exercise_planned_min. Of the five, T1D_011 is still blocked because its CGM value of 65 mg/dL triggers HYPO_CONTRA as defense in depth; the other four are unblocked.

7.5 Honest Negative Result: The Evidence Gate

Removing the evidence gate produces S5_noGate, identical to S5_full on every metric. The gate is inert because the benchmark's 13 compiled knowledge concepts are all graded A or B. We report this as a benchmark coverage limitation, not a design failure.

7.6 Inter-Seed Robustness

Point estimates at seed 7 are identical to three decimal places on every metric. The system is effectively deterministic on this benchmark.

Table 4: Case-by-case behaviour of S5_dumbPhi on the five divergent scenarios. All failures share a single root cause: the regex cannot recover `exercise_planned_min`. T1D_011 is rescued by `HYPO_CONTRA` (defense in depth).

Case	Expected Violation	Failed Field	S5_full	S5_dumbPhi
T1D_005	`EXERCISE_HYPO`	`Exercise_planned_min`	Blocked	Unblocked
T1D_011	`EXERCISE_HYPO`	`Exercise_planned_min`	Blocked	Blocked†
T1D_023	`EXERCISE_HYPO`	`Exercise_planned_min`	Blocked	Unblocked
T1D_027	`EXERCISE_HYPO`	`Exercise_planned_min`	Blocked	Unblocked
T1D_031	`EXERCISE_HYPO`	`Exercise_planned_min`	Blocked	Unblocked

Note: † Blocked by `HYPO_CONTRA` (CGM 65 < 70 mg/dL), not by the intended `EXERCISE_HYPO` rule. Defense in depth.

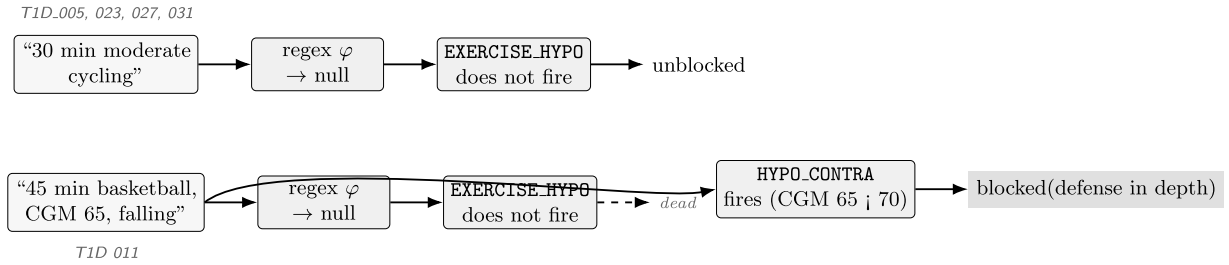

Figure 4: Causal chain of the dumbPhi failures. The regex extractor fails on the same field, `exercise_planned_min`, in all five cases. Four cases are unblocked because no other rule covers the violation. T1D_011 is blocked by a parallel chain through `HYPO_CONTRA` on its CGM value of 65 mg/dL, illustrating defense in depth.

8 Discussion

8.1 What Hard-Safety Equivalence between S4 and S5 Actually Means

S4 and S5 reach the hard-safety ceiling together. A reader who stops at this row would conclude the architectural commitments are unjustified. The right reading is the opposite: hard safety has a ceiling, and once a benchmark's hard-violation subset is fully covered by the rule library every system that contains that rule library will reach it together. The honest question is whether KRD pays off on the metrics that are not at the ceiling, and on ETC, RCB, and the API call audit it does [28,29].

When $\mathcal{R}$ is incomplete—when a hard violation exists that no rule encodes—both S4 and S5 degrade to LLM-only handling of that case. The degradation is symmetric on safety metrics but not on auditability: S5 still supplies the evidence trace from $\mathcal{K}$, and a clinician reviewing the output would see the evidence notes and the absence of a rule hit—a structured signal that the case fell outside the rule library's coverage. Validating this structural argument requires a benchmark with deliberately incomplete rules, which we flag as future work.

8.2 The Fact Extractor Is the Bottleneck

The dumbPhi result locates the bottleneck at φ. The deployment consequence: a team deploying KRD should treat φ as a first-class component subject to its own regression tests and should be unwilling to substitute a cheaper extractor without rerunning the hard-safety metrics.

8.3 The Evidence Gate Negative Result, Taken Seriously

The gate is doing exactly the work it was designed to do; it simply has no occasion to act on this benchmark. The natural follow-up benchmark is one whose concept distribution is deliberately heterogeneous in evidence grade.

8.4 Toward Layer-Wise Auditability as a Design Discipline

The present evaluation treats KRD as a standalone agent. A stronger test would embed it in a human–AI team and measure accept/edit/override rates as first-class outcomes. The three branches of δ.`compose` map naturally onto this framing: branch one corresponds to system override (block), branch three produces a clinician-editable recommendation, and the evidence trace supports the edit decision. Recent frameworks for evaluating human–AI team intelligence—including TRIAD [30] and CLIX-M [31]—provide the evaluation vocabulary that a team-level study of KRD would need. We flag this as the most important direction for future evaluation.

The fact extractor is the bottleneck. The rule layer is the safety floor. The compiled knowledge layer is the source of auditability. The decision interface is where the short-circuit lives. These four roles being explicit, separable, and independently inspectable is the property we are asking the reader to take away.

9 Limitations

The benchmark is small. Thirty-two scenarios is enough to produce clean five-way separation on strict USR but not enough to make the ETC and RCB differences statistically significant.

The language model backbone is fixed (`glm-4-flashx`, temperature 0.2). Cross-backbone robustness is untested.

The domain is single. T1D is well-suited but a single domain cannot demonstrate cross-domain transfer.

Seventeen of 32 scenarios were drafted by a language model and reviewed by the author. Any such benchmark risks being easier for models from the same family.

All labels were reviewed by the author only, not by an independent board-certified endocrinologist. This is the most important single limitation of the present evaluation.

10 Conclusion

KRD is an architecture for high-stakes clinical decision support that separates the responsibilities of fact extraction, compiled clinical knowledge, rule evaluation, and recommendation composition into four components that fail and can be inspected independently. On a 32-scenario type-1 diabetes benchmark we showed that a strict unsafe-suggestion rate stratifies five candidate systems cleanly, that hard safety has a ceiling reached together by any system whose rule library covers the violation set, that KRD leads a comparable light hybrid on evidence trace completeness and reviewer correction burden by margins that are directionally clear and borderline significant, and that KRD issues 47% fewer language model calls per benchmark pass because the short-circuit branch in its decision interface routes hard-violation cases away from the model entirely. We also reported honestly that the evidence gate is inert on this benchmark because all compiled concepts are graded A or B, and we located the operative bottleneck of the pipeline at the fact extractor through a case-by-case analysis of five regex baseline failures. The right way to read the paper is not as a claim that KRD is the right architecture for all high-stakes settings but as a claim that layer-wise auditability is a design discipline worth taking seriously, and that on the metrics this discipline was supposed to help with, in this domain, against this comparison set, it did.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Bailing Zhang: Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, project administration; Genlang Chen: Writing—review and editing, validation, investigation; All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Ethics Approval: This study does not involve human subjects or animal experiments requiring institutional ethics approval.

Conflicts of Interest: The authors declare no conflicts of interest.

Supplementary Materials: The supplementary material is available online at https://www.techscience.com/doi/10.32604/jimh.2026.084876/s1.

References

1. Rajkomar A, Dean J, Kohane I. Machine learning in medicine. N Engl J Med. 2019;380(14):1347–58.
2. Karpathy A. LLM Wiki. 2026 [cited 2026 Apr 12]. Available from: https://gist.github.com/karpathy/442a6bf555914893e9891c11519de94f/.
3. American Diabetes Association Professional Practice Committee. Standards of care in diabetes-2025. Diabetes Care. 2025;48(Suppl. 1):S1–352. doi:10.2337/dc22-ad08.
4. Battelino T, Alexander CM, Amiel SA, Arreaza-Rubin G, Beck RW, Bergenstal RM, et al. Continuous glucose monitoring and metrics for clinical trials: an international consensus statement. Lancet Diabetes Endocrinol. 2023;11(1):42–57. doi:10.1016/S2213-8587(22)00319-9.
5. Marling CR, Bunescu RC. The OhioT1DM dataset for blood glucose level prediction. In: Proceedings of the 3rd International Workshop on Knowledge Discovery in Healthcare Data (KHD@IJCAI); 2018 Jul 13; Stockholm, Sweden.
6. Singhal K, Azizi S, Tu T, Mahdavi SS, Wei J, Chung HW, et al. Large language models encode clinical knowledge. Nature. 2023;620(7972):172–80. doi:10.1038/s41586-023-06291-2.
7. Thirunavukarasu AJ, Ting DSJ, Elangovan K, Gutierrez L, Tan TF, Ting DSW. Large language models in medicine. Nat Med. 2023;29(8):1930–40. doi:10.1038/s41591-023-02448-8.
8. Garcez AD, Lamb LC. Neurosymbolic AI: the 3rd wave. Artif Intell Rev. 2023;56(11):12387–406. doi:10.1007/s10462-023-10448-w.
9. Manhaeve R, Dumancic S, Kimmig A, Demeester T, De Raedt L. DeepProbLog: neural probabilistic logic programming. Adv Neural Inf Process Syst. 2018;31:3753–63.
10. Kautz HA. The third AI summer: AAAI Robert S. Engelmore memorial lecture. AI Mag. 2022;43(1):105–25. doi:10.1002/aaai.12036.
11. Alshiekh M, Bloem R, Ehlers R, Könighofer B, Niekum S, Topcu U. Safe reinforcement learning via shielding. Proc AAAI Conf Artif Intell. 2018;32(1):2669–78. doi:10.1609/aaai.v32i1.11797.
12. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Adv Neural Inf Process Syst. 2020;33:9459–74.
13. Rebedea T, Dinu R, Sreedhar MN, Parisien C, Cohen J. NeMo Guardrails: a toolkit for controllable and safe LLM applications with programmable rails. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2023 Dec 6–10; Singapore. p. 431–45. doi:10.18653/v1/2023.emnlp-demo.40.

14. Packer C, Wooders S, Lin K, Fang V, Patil SG, Stoica I, et al. MemGPT: towards LLMs as operating systems. arXiv:2310.08560. 2023.
15. Schick T, Dwivedi-Yu J, Dessi R, Raileanu R, Lomeli M, Hambro E, et al. Toolformer: language models can teach themselves to use tools. Adv Neural Inf Process Syst. 2023;36:68539–51. doi:10.52202/075280-2997.
16. Baek J, Aji AF, Saffari A. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In: Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE); 2023 Jul 13; Toronto, ON, Canada. p. 78–106. doi:10.18653/v1/2023.nlrse-1.7.
17. Xiong G, Jin Q, Lu Z, Zhang A. Benchmarking retrieval-augmented generation for medicine. arXiv:2402.13178. 2024.
18. Panagoulias DP, Virvou M, Tsihrintzis GA. Augmenting large language models with rules for enhanced domain-specific interactions: the case of medical diagnosis. Electronics. 2024;13(2):320. doi:10.3390/electronics13020320.
19. Shortliffe EH. Computer-based medical consultations: MYCIN. Amsterdam, The Netherlands: Elsevier; 1976.
20. Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden syntax. Comput Biomed Res. 1994;27(4):291–324. doi:10.1006/cbmr.1994.1023.
21. Kawamoto K, Houlihan CA, Balas EA, Lobach DF. Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. BMJ. 2005;330(7494):765. doi:10.1136/bmj.38398.500764.8f.
22. Guyatt GH, Oxman AD, Vist GE, Kunz R, Falck-Ytter Y, Alonso-Coello P, et al. GRADE: an emerging consensus on rating quality of evidence and strength of recommendations. BMJ. 2008;336(7650):924–6. doi:10.1136/bmj.39489.470347.ad.
23. Stiell IG, Greenberg GH, McKnight RD, Nair RC, McDowell I, Worthington JR. A study to develop clinical decision rules for the use of radiography in acute ankle injuries. Ann Emerg Med. 1992;21(4):384–90. doi:10.1016/s0196-0644(05)82656-3.
24. Efron B, Tibshirani RJ. An introduction to the bootstrap. Boston, MA, USA: Springer; 1993. doi:10.1007/978-1-4899-4541-9.
25. Goddard K, Roudsari A, Wyatt JC. Automation bias: a systematic review of frequency, effect mediators, and mitigators. J Am Med Inform Assoc. 2012;19(1):121–7. doi:10.1136/amiajnl-2011-000089.
26. Team GLM, Zeng A, Xu B, Wang B, Zhang C, Yin D, et al. ChatGLM: a family of large language models from GLM-130B to GLM-4 all tools. arXiv:2406.12793. 2024.
27. Vasey B, Nagendran M, Campbell B, Clifton DA, Collins GS, Denaxas S, et al. Reporting guideline for the early stage clinical evaluation of decision support systems driven by artificial intelligence: DECIDE-AI. BMJ. 2022;377:e070904. doi:10.1136/bmj-2022-070904.
28. U.S. Food and Drug Administration. Artificial intelligence/machine learning (AI/ML)-based software as a medical device (SaMD) action plan. 2021 [cited 2026 Apr 15]. Available from: https://www.fda.gov/media/145022/download.
29. Ghassemi M, Oakden-Rayner L, Beam AL. The false hope of current approaches to explainable artificial intelligence in health care. Lancet Digit Health. 2021;3(11):e745–50. doi:10.1016/S2589-7500(21)00208-9.
30. Li J, Zhou ZC, Wang ZC, Lv H. Prioritizing human-AI collaboration in healthcare: the TRIAD framework for trustworthy governance, real-world, and integrated adaptive deployment. Mil Med Res. 2025;12:97. doi:10.1186/s40779-026-00684-w.
31. Brankovic A, Cook D, Rahman J, Delaforce A, Li J, Magrabi F, et al. Clinician-informed XAI evaluation checklist with metrics (CLIX-M) for AI-powered clinical decision support systems. npj Digit Med. 2025;8(1):364. doi:10.1038/s41746-025-01764-2.