A Two-Axis Framework for Clinical AI Architectures: Cognitive Autonomy, Neuro-Symbolic Coupling, and External Verification

Hemachandran Babu, Priyadarsini Karthik
SRM Institute of Science and Technology
Kattankulathur, India
hb6034@srmist.edu.in, priyadak@srmist.edu.in

Abstract—Clinical AI systems have evolved from simple token classifiers to retrieval-augmented, graph-grounded, and autonomous multi-agent architectures, yet task-level metrics fail to fully capture the verification effort caused by the behaviour of the system at runtime. In this paper, a two-axis evaluation framework is proposed, based on cognitive autonomy (CA) and neuro-symbolic coupling (NS). Cognitive autonomy is defined by a transparent 0–4 feature-count index and neuro-symbolic coupling is measured on a cumulative 0–4 ordinal scale. The framework is applied in an exploratory pilot to ten purposively selected systems using an independent dual-rater assessment; the sample is intended to demonstrate the rubric rather than constitute an exhaustive survey. The level of agreement at the criterion level was 85% (Cohen's κ=0.70), with quadratic weighted κ values of 0.89 for CA and 0.71 for NS. Of the five systems that achieved a high degree of cognitive autonomy, four satisfied none of the explicit neuro-symbolic criteria, while KG4Diagnosis attained the highest neuro-symbolic coupling level seen in the sample (NS=3). In order to go beyond evaluation that is limited to taxonomy, one safeguard derived from the framework—external clinical-invariant checking—was implemented in a controlled synthetic EHR-style benchmark consisting of 80 cohorts of 60 patients and 400 injected semantic faults distributed across five fault classes. All the faulty transformations were completed without any runtime exceptions, and the external checker identified 400 out of 400 faults with 0 false positives among the 80 clean runs; the mean checker latency was 2.64 ms (95th percentile 3.53 ms). These results provide proof-of-concept evidence for one framework-derived safeguard: an external contract can detect targeted semantic failures missed by an execution-success proxy. They do not establish predictive or clinical validity of the CA/NS framework.

***Index Terms*—agentic AI, clinical AI architectures, cognitive autonomy, GraphRAG, neuro-symbolic systems, architectural assessment, external verification.**

I. INTRODUCTION

The process of extracting clinical information (IE) transforms unstructured clinical narratives into entities, relations, temporal events and normalised concepts and is one of a number of clinical AI tasks—together with retrieval-augmented reasoning, knowledge-graph construction, diagnostic decision support and multi-agent clinical workflows—that are currently based on increasingly autonomous architectures. Large language models are able to carry out several of these tasks using only a small number of examples [1], whereas specific work on temporal extraction [2], multitask clinical language models [3], ClinicalBERT [4] and BioBERT [5] has set strong baseline results for various resources such as MIMIC-III [6]. The architectural context of the models, however, has changed faster than the criteria used to evaluate them.

Current retrieval augmented systems integrate dense similarity search with graph structure. ClinicalRAG obtains information from a medical knowledge graph, a knowledge base and online sources by means of a modular pipeline [7]; ARCH builds a large-scale clinical knowledge graph from aggregated narrative and codified features of health records [8]; Medical GraphRAG extends the retrieval process to include community level synthesis [9]; and general reviews explain why graph databases are becoming more relevant to clinical applications [10]. Another development can be seen in agentic systems since MDAgents dynamically recruits specialist agents [11], AgentClinic assesses tools by using agents in simulated clinical situations [12], EHRAgent generates and carries out code against tabular EHRs [13], ColaCare coordinates agents based on clinical guidelines [14] and KG4Diagnosis makes use of hierarchical routing supported by a knowledge graph [15].

Surveys currently available cover the clinical IE tasks and methods [16], ontology-based normalization [17] and the use of the extracted information for treatment modelling [18]. Recent reviews of healthcare agents also point out the rapid growth in agentic studies coupled with ongoing problems relating to validation and safety [19]. A contribution that is directly related is the seven-dimensional taxonomy developed by Vatsal, Dubey and Singh [20], in which 49 LLM-based agent studies in the field of healthcare and medicine are examined using 29 operational sub-dimensions and a labeling criterion consisting of Fully Implemented, Partially Implemented or Not Implemented. This study offers a broad characterization of capabilities and system design together with a quantitative analysis of prevalence and co-occurrence among agentic systems.

The current framework is not a substitute for the more general taxonomy but rather complementary to it; its area of analysis is more limited in that it treats runtime cognitive autonomy and explicit neuro-symbolic coupling as two distinct

architectural characteristics and applies them to static, retrieval-assisted, graph-structured and agentic clinical AI, not just to LLM-controlled agents. The question that remains unresolved in this paper is whether these two features can differ in such a way as to alter the verification requirements even when conventional task-level success signals are satisfactory. The novelty of this study is therefore not a more complete agent taxonomy but rather a focused and auditable assessment of autonomy versus symbolic coupling that is linked to the verification needs at the architectural level.

TABLE I. POSITIONING RELATIVE TO THE HEALTHCARE AGENT TAXONOMY OF VATSAL ET AL. [20]

Aspect	Vatsal et al. [20]	Present study
Scope	LLM-based agents in healthcare and medicine	Clinical AI from static extraction through agentic/tool-using systems
Corpus	49 studies	10 purposively selected systems
Structure	7 dimensions; 29 sub-dimensions	2 focused axes; 8 operational criteria
Rating	Fully / Partially / Not Implemented	CA feature-count index; NS cumulative ordinal level
Primary aim	Capability prevalence and co-occurrence mapping	Autonomy-symbolic coupling and verification burden
Boundary	LLM is the primary agent controller	Includes non-agentic and graph-structured architectures

The paper is based on three research questions. The first is: How might an architecture's runtime autonomy and the extent of its neuro-symbolic coupling through the use of explicit symbolic representation and inference be turned into reproducible descriptive measures? The second is: In the sample of architectures that has been deliberately chosen, how do cognitive autonomy and neuro-symbolic coupling vary together? The third is: What verification safeguards arise out of the identified gap, and can one of the safeguards derived from the framework detect controlled semantic failures that an execution-success proxy would accept? To answer these questions, the paper makes three contributions. The first of these is to define two observable dimensions of architecture—cognitive autonomy and neuro-symbolic coupling—and to operationalise them in different ways depending on their construct structure: treating cognitive autonomy as a transparent index of features on a 0–4 scale and neuro-symbolic coupling as a cumulative ordinal level ranging from 0 to 4. The second contribution is to build on more general healthcare-agent taxonomies such as that of Vatsal et al. [20] by applying the two-axis framework to ten systems that have been deliberately selected and which cover the range from non-agentic to agentic architectures. The third is to derive three safeguards at the level of the architecture and to carry out a clinical invariant check as a controlled proof-of-concept experiment in which external contract checks are compared with runtime completion as an internal measure of success.

II. STUDY DESIGN

A. Scope and system selection

On 19 July 2026 a literature search was carried out on PubMed in order to find systems which are representative of a given architectural approach and that have relevance to clinical AI. Four predefined Boolean search strategies were applied, these being (i) systems involving static clinical natural language processing, (ii) systems based on clinical retrieval augmented generation, (iii) knowledge graph and GraphRAG architectures and (iv) agentic and multi-agent clinical systems. The searches were limited to publications in English that had been published from 1 January 2018 up to 19 July 2026. The four searches yielded 51, 224, 349 and 607 records respectively, giving a total of 1,231 records prior to deduplication; after the elimination of 55 duplicate records, 1,176 unique records constituted the candidate pool for purposive architectural sampling rather than forming an exhaustively screened systematic-review corpus. The complete Boolean query result counts and the number of records that were duplicated were kept in a structured search log.

Ten systems were selected on purpose from the combined candidate list produced by the PubMed searches and the additional targeted searches, using four criteria: having a clinical or biomedical application; having a primary peer-reviewed technical publication; being architecturally distinct; and having enough implementation details so that the proposed rubric could be applied. The sample was intended to include examples of static, retrieval-assisted, graph-structured and agentic clinical AI architectures, not to provide an estimate of how common any particular pattern is in the literature. Accordingly, the ten-system assessment is an exploratory pilot intended to demonstrate the application and reproducibility of the rubric across contrasting architectures; it is not an exhaustive survey or an estimate of the prevalence of CA/NS configurations in clinical AI. Studies that focused only on imaging, those that focused only on genomics, general-purpose agent frameworks that had not been evaluated in a clinical context, and publications that did not contain sufficient information about the architecture were all excluded. Five of the selected systems—ClinicalRAG, Medical GraphRAG, KG4Diagnosis, MDAgents and ColaCare—could not be found through the PubMed searches and were therefore located using targeted searches of the ACL Anthology, the ACM Digital Library, the NeurIPS proceedings and PMLR, each of them being checked against their official peer-reviewed publication. Two more systems—EHRAgent and AgentClinic—were obtained via the PubMed searches and were then independently verified against their official primary publications.

B. Illustrative architectural families

To help with the presentation, the systems chosen were divided into four illustrative architectural families. Family A consisted of static extraction systems, Family B included retrieval-assisted architectures, Family C comprised graph-structured representation or retrieval systems, and Family D covered agentic or tool-using architectures. These groups are merely descriptive and should not be interpreted as stages in a process of maturation or as part of a one-dimensional hierarchy; the structure of representation and the degree of runtime autonomy can each vary independently, for instance, a graph-structured architecture can have only limited runtime autonomy, while an agentic architecture can function with minimal explicit symbolic grounding. For this reason, the comparative analysis in the study is based on the two independent dimensions introduced in Section III rather than on the order of these architectural families.

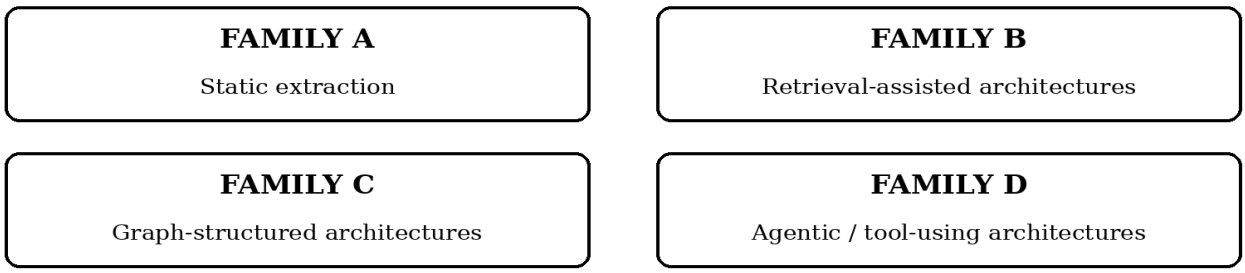


Fig. 1. Illustrative architectural families used to organize the representative systems. The families are descriptive categories and do not constitute a maturity hierarchy.

C. Unit of analysis

The unit of analysis is the specific computational architecture or evaluated configuration described in each cited primary publication, not the publication, dataset, benchmark, or product category as a whole. Scores refer only to runtime or symbolic behavior documented for that configuration. Thus, ARCH is assessed as the knowledge-graph-construction architecture reported in [8], whereas for AgentClinic the scored unit is the doctor-agent behavior within the standard interactive workflow described in [12], not the benchmark infrastructure or the supporting patient, measurement, and moderator agents. The framework therefore compares documented architectural properties across heterogeneous clinical tasks rather than treating the ten entries as equivalent products or head-to-head systems.

III. Two-axis assessment framework

A. Cognitive autonomy

Cognitive autonomy is defined by a transparent index based on the number of observable instances of runtime discretion. It is obtained from four binary criteria. For CA1 (dynamic task decomposition or planning) the criterion is satisfied when the system breaks down the case or objective into subgoals, steps or a plan during runtime rather than merely adhering to a fixed predefined sequence. CA2 (runtime selection or routing) is met when the system chooses among tools, agents, data sources, models or reasoning routes on the basis of the current case at runtime. CA3 (iterative execution with intermediate state) is satisfied when the system carries out several reasoning or action cycles and makes use of the intermediate outputs or execution results in later steps. CA4 (self-evaluation and autonomous correction) is met when the system assesses its own output, identifies a failure or inadequacy and then alters its plan, action, code, retrieval or answer without any human causing the correction. Each criterion is rated as Yes, No or Unclear in relation to the primary source, with Unclear being used only in those cases where the source actually does not give enough information to make a decision. Each criterion that is satisfied awards one point, so that CA values range from 0 to 4. Equal weighting is intentionally employed in order to maintain interpretability, reproducibility and auditability; it does not mean that the four capabilities are equally important in a clinical context or that the resulting values form an interval-scale measure.

B. Neuro-symbolic coupling

Neuro-symbolic coupling is defined by a cumulative ordinal index which describes a step-by-step strengthening of the integration of explicit symbolic knowledge into the system's computation. The four criteria correspond to increasing degrees of coupling. For NS1 (the presence of explicit machine-readable symbolic knowledge) to be satisfied, the architecture must make use of an explicit ontology, a controlled vocabulary, a knowledge graph, rules, codes, a schema or some other machine-readable symbolic resource. Ordinary storage, tables or API schemas do not count unless they include domain-semantic concepts, relations or constraints that have a real impact on the way the system processes information. NS2 (the integration of symbolic representations into computation) is achieved when symbolic entities, relations, codes or graph structures are incorporated into the representation, retrieval, model input or internal processing, rather than being used merely for annotation or evaluation. NS3 (symbolic knowledge guiding or constraining processing) is fulfilled when symbolic knowledge directly guides or constrains retrieval, reasoning, generation, routing or the selection of candidates in such a way that altering the symbolic structure would result in a change to the system's processing path or to the set of allowable outputs. NS4 (the performance of explicit symbolic inference, validation or consistency checking) is met when the system carries out inference or validation based on graphs, rules, logic, constraints, provenance or ontology in order to verify or derive its outputs, as opposed to simply carrying out ordinary LLM self-critique or similarity retrieval. Each criterion is assessed as Yes, No or Unclear with respect to the primary source using the same evidentiary standard as that applied to the cognitive-autonomy criteria. The NS criteria are to be interpreted cumulatively rather than as four independent statistical features: the system is given the highest level for which that criterion and all the previous levels are satisfied. Therefore, an NS score of 3 means that NS1 through NS3 are satisfied but NS4 is not. The numerical values indicate an ordinal progression in the strength of coupling and must not be understood as representing equally spaced measurement units.

C. Scoring rule and interpretation

For a system s, cognitive autonomy is calculated as

$$CA(s) = CA1 + CA2 + CA3 + CA4$$

NS(s) = highest cumulative NS level reached (0 to 4)

With regard to CA, one point is awarded for each criterion that is satisfied. For NS, the assigned value is the highest cumulative level attained after all the previous NS criteria have been satisfied. Any criteria that are unclear must be resolved or recorded as uncertainty prior to the final assignment. CA is thus a transparent index based on the number of features, while NS is an ordinal measure of cumulative coupling. Neither of the two values is meant to serve as a measure of clinical performance, as a psychometrically validated latent scale, or as evidence of clinical safety. Their function is to offer reproducible descriptions of the architecture which can be directly traced to the features described in the main publication.

As a worked example, KG4Diagnosis satisfies CA1 through CA3 (triage-based decomposition, specialist routing and iterative agent exchange) but does not meet CA4 since no autonomous re-evaluation loop is mentioned, which results in a CA rating of 3. It also fulfils NS1 to NS3 (as it makes use of an explicit knowledge graph that is incorporated into and guides specialist reasoning) but does not satisfy NS4 because the cited evidence only refers to the construction of the graph, not to a runtime inference or validation step, thus earning a NS score of 3. Evidence supporting each criterion is recorded together with the location in the original source, in accordance with the independent dual-rater assessment outlined in Section VIII.

D. Rubric development and construct status

The criteria were selected to represent two distinct, observable architectural constructs. CA1–CA4 capture distinct forms of runtime discretion—planning, case-dependent routing, iterative use of intermediate state, and autonomous correction. CA is deliberately an unweighted feature count because no empirical basis currently supports differential clinical-risk weights; equal weighting favors transparency and auditability and does not imply equal clinical importance. NS is cumulative because each higher level presupposes the earlier one: symbolic knowledge must exist before computational integration, integration before it can guide or constrain processing, and explicit inference or validation forms the highest defined level. These choices provide face and content rationale for an exploratory architectural rubric; dual-rater agreement assesses scoring reproducibility, not construct or predictive validity.

TABLE II. OPERATIONAL DEFINITIONS OF THE TWO ASSESSMENT AXES

Criterion	Cognitive autonomy	Neuro-symbolic coupling
1	Dynamic task decomposition or planning	Explicit machine-readable symbolic knowledge
2	Runtime selection or routing	Symbolic representation integrated into computation
3	Iterative execution with intermediate state	Symbolic knowledge guides or constrains processing
4	Self-evaluation and autonomous correction	Explicit symbolic inference, validation or consistency checking

IV. APPLICATION TO REPRESENTATIVE SYSTEMS

A. Score assignment

Each system was independently scored by two raters against the finalized CA1–CA4/NS1–NS4 rubric using the primary publication as evidence. MT-clinical BERT was scored CA=0, NS=0. CLEAR and ClinicalRAG were each scored CA=0, NS=3; their entity- or graph-based retrieval shapes the evidence passed downstream, but the publications do not document runtime planning, routing, iteration, or self-correction. ClinicalRAG is therefore scored from documented runtime behavior rather than from its “multi-agent pipeline” label. ARCH was scored CA=0, NS=2: its knowledge graph satisfies NS1–NS2, but the cited implementation describes offline graph construction using co-occurrence embeddings, cosine similarity, and sparse regression rather than runtime symbolic guidance, inference, validation, or consistency checking, so NS3–NS4 are not met. Medical GraphRAG was scored CA=1, NS=3. EHRAgent was scored CA=4, NS=0 because it plans, routes, iterates, and revises generated code after execution feedback, while its documented tabular schema and code-execution constraints do not meet the rubric’s threshold for domain-semantic symbolic knowledge. MDAgents, ColaCare, and the assessed AgentClinic doctor-agent configuration were each scored CA=3, NS=0: they plan, route, and iterate, but the retained evidence does not establish a distinct autonomous correction loop satisfying CA4, and their moderation, text-retrieval, tool-use, or database mechanisms do not constitute an explicit domain-semantic symbolic resource satisfying NS1. KG4Diagnosis was scored CA=3, NS=3; its hierarchical routing is knowledge-graph guided, but a second source audit confirmed NS4=No because the cited evidence concerns entity extraction and relationship reconstruction during graph preparation rather than runtime symbolic inference, validation, or consistency checking.

Twelve of the eighty initial criterion decisions (15%) differed between the two raters and were resolved through joint re-examination of the primary source. Criterion-level exact agreement was 85% (Cohen’s κ=0.70). Rater A assigned 29 Yes and 51 No decisions, whereas Rater B assigned 39 Yes and 41 No decisions; neither rater used Unclear. For CA, the Yes/No distributions were 15/25 for Rater A and 19/21 for Rater B; for NS they were 14/26 and 20/20, respectively. Axis-level quadratic-weighted Cohen’s κ was 0.89 for CA and 0.71 for NS. Because the sample contains only ten systems, these estimates are exploratory; 95% percentile bootstrap confidence intervals of [0.63, 1.00] and [0.31, 0.95] were obtained from 5,000 system-level resamples with a fixed random seed. These agreement statistics refer to the original independent ratings. Four disputed criteria (ARCH NS3 and NS4, AgentClinic CA4, and MDAgents CA4) subsequently underwent a second independent source audit and were assigned No by both raters; KG4Diagnosis NS4 was re-audited using the same evidentiary standard and was also finalized as No. The final scores reported in the paper reflect these source-audit decisions. Appendix A reports the initial A/B positions, retained evidence locations, and final decisions for the boundary criteria CA4 and NS4.

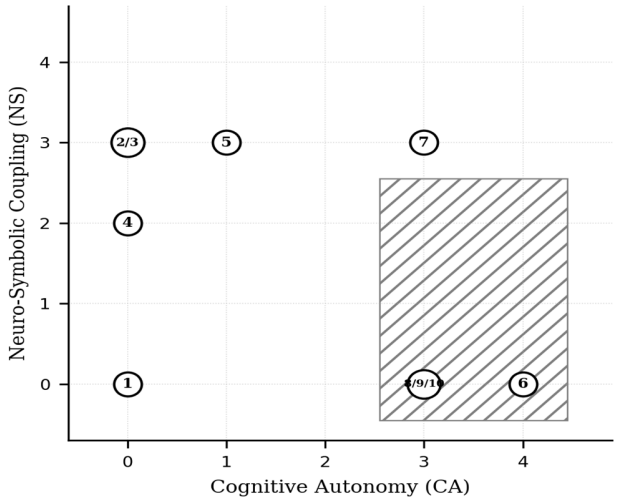


Fig. 2. Dual-rater consensus taxonomy map. The hatched region denotes high autonomy without the highest neuro-symbolic coupling level reached in the sample (NS=3).

1 MT-clinical BERT, 2 CLEAR, 3 ClinicalRAG, 4 ARCH, 5 Medical GraphRAG, 6 EHRAgent, 7 KG4Diagnosis, 8 MDAgents, 9 ColaCare, 10 AgentClinic.

TABLE III. APPLICATION OF THE RUBRIC TO TEN REPRESENTATIVE SYSTEMS

System	CA	NS	Architectural role / assessed configuration
MT-clinical BERT [3]	0	0	Static clinical information extraction
CLEAR [21]	0	3	Retrieval-assisted extraction
ClinicalRAG [7]	0	3	Graph-augmented retrieval
ARCH [8]	0	2	Clinical knowledge-graph-construction architecture
Medical GraphRAG [9]	1	3	Graph retrieval (community-level)
EHRAgent [13]	4	0	EHR tabular reasoning agent
KG4Diagnosis [15]	3	3	Multi-agent diagnosis (graph-guided)
MDAgents [11]	3	0	Multi-agent decision-making
ColaCare [14]	3	0	Multi-agent EHR prediction
AgentClinic [12]	3	0	Doctor-agent configuration in interactive benchmark

B. Observed architectural pattern

The map shown in Fig. 2 distinguishes between two types of architectural variation. The systems in Family C enhance explicit grounding while still maintaining a limited amount of runtime branching, whereas in the case of the Family D systems coupling is mostly absent rather than only partially present. The key point is not that all autonomous systems are generally only weakly grounded, but that four of the five systems which achieved high cognitive autonomy (EHRAgent, MDAgents, ColaCare and AgentClinic) had NS=0: none of them reached the first level of explicit symbolic grounding. KG4Diagnosis is the only exception, attaining CA=3 and NS=3 by combining hierarchical agent routing with knowledge graph guided reasoning, even though it does not meet its NS4 criterion (explicit symbolic inference or validation). The graph is created by means of entity and relationship extraction and is not used during a runtime inference or validation step. The fact that it is included in the sample shows that it is possible to have both high autonomy and considerable symbolic grounding even though no system in the sample reached the strictest level of coupling. The remaining four systems indicate that partial grounding is not the normal result of increasing autonomy.

C. Threshold and documentation sensitivity

The map makes a specific assertion about the rubric's highest symbolic-coupling level rather than a general claim that autonomous systems lack symbolic grounding. With strong coupling defined as NS≥3, CLEAR, ClinicalRAG, Medical GraphRAG, and KG4Diagnosis meet the threshold, and KG4Diagnosis is the only one of these that also has high cognitive autonomy. At the stricter NS=4 level, no sampled system qualifies. NS4 denotes explicit symbolic inference, validation, or consistency checking within the architecture; it does not require a separate external component capable of overriding the system output. ARCH satisfies NS1–NS2 but not NS3 or NS4 because the documented graph is computationally represented but its construction does not guide runtime processing or perform runtime symbolic inference. KG4Diagnosis satisfies NS1–NS3 but not NS4 because its graph guides specialist reasoning while the cited evidence for the fourth criterion concerns graph preparation rather than runtime inference, validation, or consistency checking. The qualitative low-coupling pattern was robust to a ±1-unit sensitivity check, although exact counts were less stable: changing an NS=0 score to NS=1 alters the zero-coupling count, reducing a CA=3 system to CA=2 changes membership of the high-autonomy subset, lowering KG4Diagnosis from NS=3 to NS=2 removes the sole high-autonomy example at the highest observed coupling, and increasing any NS=3 system to NS=4 changes the finding that no sampled system reaches the maximum level. Leave-one-out exclusion of KG4Diagnosis preserves the broader low-coupling pattern but removes that single high-autonomy/highest-coupling example.

Three sensitivity checks were applied to the consensus scores: threshold variation, leave-one-out analysis, and a one-unit score/level perturbation. The qualitative low-coupling pattern persisted when the high-autonomy threshold was set at either CA=3 or CA=4, whereas exact counts remained sensitive to individual systems and score boundaries. Because the rubric depends on published architecture descriptions, an undocumented feature cannot be credited; unresolved features should be recorded as unknown rather than assumed absent.

V. CROSS-ARCHITECTURE RESULTS

A. Proxy success signals change with architecture

Family A usually considers an exact match to be a success. While this is helpful in the case of a stable labeling task, it may fail to identify downstream semantic errors; for instance, a correctly detected diagnosis can have the negation or temporality lost when it is passed on to a separate normalizer [2], [17]. Family B includes retrieval metrics such as Recall@k, but a high retrieval score does not ensure that all the parts of a compound clinical rule are retrieved together. A fixed chunk boundary can cause a qualifying condition to be separated from

its exception, leaving the generator with an incomplete yet internally coherent context.

Family C substitutes linear proximity with graph topology. By using a graph structure, systems are able to link evidence to specific relationships and paths and thus improve traceability as compared to unstructured retrieval [8], [10]. Although local traversal may fail to identify evidence that is spread out over weakly connected areas of the graph, global community summarization can lead to increased latency and a loss of abstraction [9]. HyperGraphRAG proves that n-ary relations can be represented by hyperedges rather than being broken down into separate binary triples [22], and HypKG shows that it is feasible to use contextualized hypergraph representations in healthcare [23]. These methods are relevant since compound clinical conditions are generally relational units rather than separate facts.

Family D presents success signals that are a long way from being clinically accurate. Although EHRAgent can modify the code until it runs, a successful execution only confirms correct syntax and that the program has completed running, not that the intended clinical cohort has been preserved [13]. MDAgents are able to achieve agreement among the specialists who have been recruited, yet this consensus can still depend on an incorrect earlier routing decision [11]. ColaCare is capable of obtaining authoritative guidance, but the query used to retrieve it may carry over an initial framing from the developing dialogue [14]. The main value of AgentClinic lies in the fact that it assesses tool use and bias within a simulated interactive environment rather than depending solely on static question answering [12].

TABLE IV. CROSS-ARCHITECTURE VERIFICATION GAP IDENTIFIED BY THE FRAMEWORK

Fam.	Typical success proxy	Unobserved failure class
A	Entity/token match	Negation, temporality or normalization drift after extraction
B	Retrieval recall / answer accuracy	Separated conditions and incomplete evidence context
C	Graph path or grounded generation	Topology limits, global synthesis loss, stale ontology links
D	Execution, task completion or consensus	Clinically incorrect action that is internally successful

B. Expected verification burden with autonomy

All architectural families exhibit a proxy-correctness gap: token match, retrieved evidence, execution completion, or internal consensus can be legitimate engineering signals without establishing clinical correctness outside the conditions they directly measure. As autonomy increases, longer chains of intermediate decisions create more opportunities for an undetected proxy error to propagate. The framework therefore hypothesizes that high cognitive autonomy—especially when explicit neuro-symbolic coupling is limited—creates a greater need for external verification; this is a prospective engineering hypothesis for the validation protocol in Section VIII.C, not an observed risk relationship. The synthetic benchmark in Section VII mirrors one motivating pattern from EHRAgent (CA=4, NS=0): execution completion is treated as the internal success signal while an independent contract checks semantic output properties. This analogy does not establish an association between CA/NS scores and failure probability.

VI. EXTERNAL VERIFICATION ARCHITECTURE

A. Design principle

The verification part should not rely entirely on the same reasonings that were used to produce the candidate output. This is in accordance with the software engineering principle of Design by Contract, whereby the preconditions, postconditions and invariants are specified independently of the component that is carrying out the execution [24]. In the case of clinical agents, the equivalent requirement is to establish the constraints before reasoning starts and to evaluate them outside the agent's self-assessment loop.

B. Three derived safeguards

Clinical invariant checking addresses code-executing agents. Rather than interpreting generated code, the checker compares the resulting data state and output properties against predefined constraints. Examples include preservation of patient identifiers, non-negative medication doses, permissible temporal ordering, and cohort-membership requirements derived from the clinical query. A transformation that executes successfully but violates an invariant is blocked or escalated.

Route-divergence testing addresses hierarchical specialist routing. Ambiguous cases are evaluated through at least one alternative route, and material disagreement between routes triggers escalation rather than allowing a downstream moderator to validate only the initially selected path. This converts routing from a latent decision into an explicitly tested architectural branch.

Evidence-query freezing addresses dialogue-driven retrieval. Initial retrieval queries are generated once from the raw patient presentation before multi-agent debate. Agents may discuss the retrieved evidence, but subsequent dialogue cannot silently redefine the evidence request. A controlled second retrieval is permitted only through a logged rule that records why additional evidence is required.

C. Implementation and evaluation implications

The three safeguards establish specific engineering objectives. The checking of invariants can be assessed by looking at the fault-detection rate, the false-block rate, and the extra latency; route-divergence testing can be measured by examining the escalation sensitivity on deliberately ambiguous cases; and evidence-query freezing can be checked by referring to the logs which compare the original input query set with the subsequent evidence requests. Section VII puts into practice the first of these safeguards as a controlled proof-of-concept experiment, while route-divergence testing and evidence-query freezing stay as proposed tests and are not given as experimentally validated controls in this study.

The controls are intentionally designed to be modular and can be carried out using rules, schema validators, ontology constraints, database assertions or human-review thresholds. Their aim is not to get rid of language models from clinical AI systems but rather to separate flexible reasoning from the power

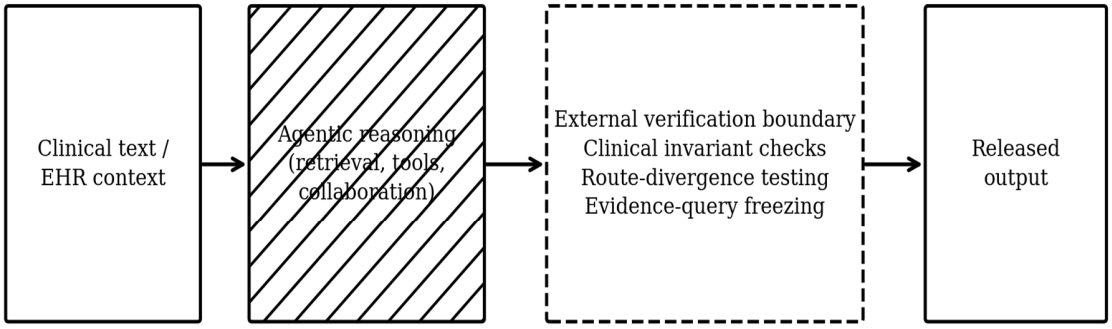


Fig. 3. Proposed placement of external verification controls outside the agentic reasoning substrate

to release an unchecked result. NS=4 means that there has been explicit symbolic inference, validation or consistency checking within the framework; it does not in itself mean that a separate external component can override or block the system's output.

VII. Proof-of-concept invariant checking experiment

Clinical invariant checking was implemented as a controlled proof-of-concept so that at least one framework-derived safeguard was evaluated rather than remaining a design proposal. This experiment evaluates the safeguard independently of the ten-system scoring exercise: it asks whether an external contract can reject semantically invalid EHR-style transformation outputs even when the transformation completes without a runtime exception. It is an engineering enforcement test, not a clinical-efficacy study and not a validation of the complete CA/NS construct or of any association between CA/NS score magnitude and fault probability.

A. Synthetic benchmark and query contract

Synthetic cohorts independently generated with a fixed random seed (20260901) consisted of 60 patients each, giving a total of 4,800 patient-level records. Each cohort included four deliberately included states: no medication record, medication only before the index date, medication only after the index date, and medication both before and after the index date. The benchmark query had to transform the data so as to produce exactly one row for each eligible patient, keep those patients who had no qualifying post-index exposure, and, in the case where such exposure did exist, return the earliest medication event that occurred on or after the patient's individual index date. Real patient data were not used.

The external contract verified six predefined properties of the output: preservation of cohort membership, one row for each patient, integrity of the link between patient and medication, correct temporal ordering of the exposure with respect to the index date, a positive numeric medication dose in cases where exposure was present, and the absence of medication values when the exposure flag was false. These checks assess the semantics of the transformation rather than simply verifying that the transformation code has completed running.

B. Fault injection and evaluation protocol

Once per cohort, five fault categories were introduced, resulting in an additional 400 faulty runs together with 80 clean reference runs. The faults were: INNER_JOIN_DROP, which causes patients without a qualifying post-index exposure to be omitted; DUPLICATE_PATIENT, which results in one patient output being duplicated; TEMPORAL_LEAK, which substitutes a real pre-index medication event for a valid post-index event; NEGATIVE_DOSE, which alters one of the exposed medication doses to a negative value; and WRONG_PATIENT_LINK, which associates an exposure with a different patient identifier. The process of constructing the cohort ensured that each injected fault led to a material contract violation.

For every run, runtime completion was used as a proxy for internal execution success, and the same output was subsequently checked by the external contract checker. A detection was considered to have occurred if there was at least one contract violation. False blocking was calculated based on the 80 clean runs. Checker latency was determined by using a high-resolution monotonic timer during the contract evaluation; the latency figures obtained are specific to the implementation and platform and are given solely to indicate the order of magnitude of the extra checking cost.

C. Results

TABLE V. Proof-of-concept invariant-checking results

Condition	Runs	Executed	Flagged
INNER_JOIN_DROP	80	80/80	80/80
DUPLICATE_PATIENT	80	80/80	80/80
TEMPORAL_LEAK	80	80/80	80/80
NEGATIVE_DOSE	80	80/80	80/80
WRONG_PATIENT_LINK	80	80/80	80/80
Clean reference	80	80/80	0/80

The 480 transformations were carried out without any runtime exceptions occurring, and therefore the execution-success proxy accepted all 400 of the intentionally defective outputs. The external checker identified all 400 of the faulty

runs (100%), comprising 80 out of the 80 cases for each fault category, while it found none of the 80 clean runs (a false-block rate of 0%). For the 480 runs in total, the average checker latency was 2.64 ms, the median latency was 2.41 ms and the 95th percentile latency was 3.53 ms; the highest latency observed for the checker was 17.49 ms. Since the fault types had been deliberately chosen to target specific invariants, the 100% detection rate must be seen as proof that the implemented contract enforces those known properties under controlled conditions and not as an indication of the system's general clinical fault-detection accuracy.

D. Scope and reproducibility

The experiment provides empirical support for the operation of one framework-derived safeguard: an external verification layer can reject targeted semantic failures that are invisible to runtime completion alone. It does not show that CA/NS scores predict clinical failure rates, estimate an association between score magnitude and fault probability, test LLM-generated code, or assess unseen fault categories, real EHR heterogeneity, route-divergence testing, or evidence-query freezing. Because the five injected faults were contract-targeted by construction, the 100% detection rate must not be extrapolated to ambiguous, emergent, or previously unseen clinical failure modes. Future evaluation should use independently generated fault sets, realistic EHR heterogeneity, and failures not used to define the checker. The fixed-seed benchmark generator, invariant checker, per-run results, and aggregate summary are retained by the authors and are available on reasonable request.

VIII. Reproducibility and dual-rater assessment protocol

A. Independent scoring

A reproducible application should retain an evidence packet for each system containing the exact publication version and the page, figure, algorithm, or implementation statement supporting every criterion. Features are marked present, absent, or unknown; incomplete reporting should not automatically be converted into absence. Two raters scored the systems independently using the finalized rubric and evidence rules in Section III, without viewing or discussing each other's initial decisions. The raters had backgrounds in clinical medicine and computer science, respectively. Agreement is reported at the criterion level with exact agreement and Cohen's κ and at the axis level with quadratic-weighted Cohen's κ; disagreements are resolved by returning to the primary source rather than averaging scores. Appendix A reports the retained source locations, evidence summaries, initial A/B decisions, and final decisions for CA4 and NS4, the two boundary criteria. The complete 80-decision evidence matrix and original rater files are retained by the authors and are available on reasonable request.

B. Controlled architectural ablation

Apart from the proof-of-concept contract experiment described in Section VII, two complementary controlled ablation studies are required in order to determine whether the taxonomy can predict the effective intervention. In the first of these, the architecture, prompt schema, retrieval index, ontology version and the validation rules should remain unchanged while the underlying model is substituted among different model families; the transitions in per-category errors would then reveal model-sensitive failures. In the second study, the model itself is kept constant and one architecture component is altered at a time, for example structure-aware segmentation, graph or hypergraph retrieval, invariant checking, alternative-route testing or frozen evidence queries. Any failure that diminishes when the targeted ablation is applied would be regarded as architecture-sensitive; a failure affected by both interventions would be classified as hybrid.

Validation ought to provide more than just the overall accuracy of tasks. It should include category-level error counts, confidence intervals, false-block and escalation rates, as well as the extra latency and implementation cost. The framework would only receive stronger predictive support if the information in its feature records enables it to predict where failures will occur and which intervention is appropriate; merely having a high or low axis score is not proof of clinical efficacy. Release of the evidence packets, benchmark manifests and ablation configurations would enable independent replication even if the proprietary models cannot be shared.

C. Predictive validation protocol

Predictive validation must be separated from the development of the rubric in a prospective manner. In a future study, the CA/NS rubric should be fixed before the data on outcomes is collected, a larger sample of architectures should then be assessed without having access to the failure labels, and the system-case failures should be independently annotated using the execution logs or the benchmark cases. The main analyses should examine whether the CA/NS configurations can predict the pre-specified failure classes beyond architectural family and task type using the leave-one-system-out validation and uncertainty intervals. The secondary analyses should look at whether the safeguards chosen from the framework reduce the predicted failure classes without incurring unacceptable false-block, escalation or latency costs. This approach would avoid the circularity that exists between the definition of the rubric and the labelling of the outcomes and would provide the evidence necessary to establish whether the two axes have predictive utility or merely show descriptive reproducibility.

IX. Threats to validity

The framework has seven limitations. First, the ten-system assessment is a purposive exploratory pilot designed to introduce and test the reproducibility of the rubric, not an exhaustive survey or prevalence estimate. Second, criterion-level scores were assigned independently by two raters; exact agreement was 85% (Cohen's $\kappa = 0.70$) and axis-level quadratic weighted κ was 0.89 for cognitive autonomy and 0.71 for neuro-symbolic coupling. Twelve of eighty initial decisions differed, and given the small sample these reliability estimates should be interpreted cautiously. Third, architectural papers differ in reporting detail, so an unreported capability may be scored as absent when the source does not document it. Fourth, the two axes simplify a multidimensional design space and do not directly measure privacy, latency, usability, calibration, clinical

effectiveness, or deployment risk. Fifth, the search was restricted to PubMed with targeted supplementation from the ACL Anthology, ACM Digital Library, NeurIPS proceedings, and PMLR. Sixth, the assessed units span models, retrieval pipelines, graph-construction architectures, and agent configurations across heterogeneous tasks; the unit-of-analysis rule in Section II.C limits each score to the documented configuration but does not make the systems head-to-head comparable. Seventh, the invariant-checking experiment uses synthetic EHR-style cohorts and five pre-specified faults chosen to violate known invariants. It therefore demonstrates contract enforcement under controlled conditions, not generalization to real clinical data, ambiguous errors, or previously unseen failure modes.

These limitations constrain both the architecture map and the proof-of-concept results. The ten-system map should generate testable engineering questions rather than rank products or infer deployment safety, and CA/NS should currently be interpreted as auditable architectural descriptors rather than risk scores. Section VII validates only the operation of one derived safeguard against targeted contract violations; route-divergence testing and evidence-query freezing remain untested in this study. Predictive support for the framework would require a larger, task-stratified architecture sample with failure labels generated independently of the rubric, followed by prospective evaluation of whether CA/NS configurations anticipate failure classes or identify safeguards that reduce those failures without unacceptable false blocking, escalation, or latency.

X. Conclusion

Today's clinical AI architectures are not distinguished merely on the basis of the model used to label text. Instead, aspects such as retrieval, graph traversal, code execution, planning, and multi-agent collaboration affect both the range of things a system can do and the aspects that its performance metrics are unable to detect. The framework in question makes this change explicit by means of a transparent cognitive-autonomy feature index and a cumulative neuro-symbolic coupling level that is derived from observable architectural features. When applied as an exploratory pilot to ten purposively selected systems through an independent dual-rater evaluation, it reveals a sample-level verification gap in that of the five systems which achieve a high degree of cognitive autonomy, four obtain a score of zero on all four neuro-symbolic criteria, while KG4Diagnosis achieves the highest coupling level seen in the sample (NS=3), even though it does not reach the strictest level specified in the rubric.

The framework also includes three external safeguards: clinical invariant checking, route-divergence testing and evidence-query freezing. When a controlled proof-of-concept version of the invariant checking was carried out, it became clear why the difference between internal and external success signals is important: all 400 of the injected semantic faults were completed without any runtime exceptions, but the external checker rejected all 400 of them while generating no false blocks on 80 runs with clean inputs. This outcome confirms that the pre-specified contract properties are enforced within the synthetic benchmark, not that the entire framework has clinical validity.

The main outstanding issue is predictive validity. In order to carry out a definite test, a larger sample of architectures organised by task type would need to be used, one that is scored using the same pre-determined rubric, together with independently compiled failure labels which are not obtained from the CA/NS criteria themselves. The analysis must examine whether CA/NS configurations are linked to particular failure types after taking into account task family, model family and the degree of reporting completeness, and also whether the safeguards chosen from the framework reduce those failures without causing unacceptable false blocking, escalation or latency. The validation approach should also keep architectural scoring separate from the annotation of outcomes: one group should assign the CA/NS values based on the system descriptions, while a different group should label the various failure categories from the execution logs or from the benchmark cases. The associations should be estimated using uncertainty intervals and subjected to leave-one-system-out verification, and then compared with simpler baseline approaches such as using the architectural family alone. The framework will only gain predictive support if CA/NS configurations consistently lead to a better anticipation of failure types or help in identifying the safeguard that reduces them. Only in this way can the evaluation tell apart construct reproducibility from predictive utility and so it represents the next essential step before the framework can be considered a validated assessment method. Until such evidence has been obtained, CA/NS should be regarded as an exploratory descriptive architectural rubric and auditable architectural descriptors rather than as risk scores or predictors of deployment failure. The key engineering implication is that as autonomy is increased, so too should verification mechanisms have explicit, external and auditable acceptance criteria.

A further implication is that future evaluations should treat verification capability as a system-level property rather than as an afterthought added once task accuracy has been measured. Architectures with similar predictive performance may expose very different verification requirements depending on how many autonomous decisions they make, how strongly those decisions are constrained by explicit symbolic knowledge and whether an independent mechanism can interrupt or reject an internally successful result. For this reason, future benchmark design should report not only conventional task metrics but also the location of decision points, the type of internal success signal used at each stage, the availability of external checks and the consequences of a failed check. Such reporting would make it possible to compare architectures on both performance and verifiability and would help determine whether additional autonomy is accompanied by a proportionate increase in assurance. In practical deployment, this distinction may be especially important because the safest architecture may not always be the one with the highest autonomy or the strongest symbolic component in isolation, but the one in which autonomous actions, symbolic constraints and independent verification mechanisms are aligned with the clinical consequences of error.

STATEMENTS AND DECLARATIONS

Author Contributions. Hemachandran Babu: conceptualisation of the framework, literature search, evidence compilation and preparation of the original manuscript draft. Priyadarsini Karthik: supervision, critical review of the framework and manuscript review and editing. Both authors reviewed the manuscript, provided intellectual input and approved the final version for submission.

Funding. The authors did not receive support from any organisation for the submitted work.

Competing Interests. The authors declare no competing interests.

Ethics Approval. No patient participants, clinical intervention or new patient level data collection were involved. The two raters provided methodological assessments of published system descriptions and agreed to participate in the scoring exercise.

Data and Code Availability. Appendix A reports the initial rater positions, retained evidence locations, and final decisions for CA4 and NS4. The complete 80-decision evidence matrix and original rater scoring files are available from the corresponding author on reasonable request. The fixed-seed benchmark generator, invariant-checking code, per-run results, and aggregate summary are likewise available on reasonable request.

Use of Generative AI and AI-Assisted Technologies. The authors used Claude (Anthropic) and ChatGPT (OpenAI) for search-query formulation, source-verification documentation, rubric refinement, supporting assessment materials, methodological consistency checks, and language drafting and revision. PubMed searches were executed and verified by the authors. All sources, numeric results, scoring decisions, statistics, and AI-assisted text were independently reviewed and approved by the authors, who take responsibility for the final manuscript.

REFERENCES

[1] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim and D. Sontag, "Large language models are few-shot clinical information extractors," in Proc. 2022 Conf. Empirical Methods in Natural Language Processing (EMNLP), 2022, pp. 1998–2022.

[2] G. Moharasan and T. B. Ho, "Extraction of temporal information from clinical narratives," J. Healthcare Informatics Research, vol. 3, no. 2, pp. 220–244, 2019.

[3] A. Mulyar, O. Uzuner and B. McInnes, "MT-clinical BERT: Scaling clinical information extraction with multitask learning," J. American Medical Informatics Association, vol. 28, no. 10, pp. 2108–2115, 2021, doi: 10.1093/jamia/ocab126.

[4] K. Huang, J. Altosaar and R. Ranganath, "ClinicalBERT: Modeling clinical notes and predicting hospital readmission," arXiv:1904.05342, 2019.

[5] J. Lee et al., "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," Bioinformatics, vol. 36, no. 4, pp. 1234–1240, 2020, doi: 10.1093/bioinformatics/btz682.

[6] A. E. W. Johnson et al., "MIMIC-III, a freely accessible critical care database," Scientific Data, vol. 3, Art. no. 160035, 2016, doi: 10.1038/sdata.2016.35.

[7] Y. Lu, X. Zhao and J. Wang, "ClinicalRAG: Enhancing clinical decision support through heterogeneous knowledge retrieval," in Proc. 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024), Bangkok, Thailand, 2024, pp. 64–68, doi: 10.18653/v1/2024.knowllm-1.6.

[8] Z. Gan et al., "ARCH: Large-scale knowledge graph via aggregated narrative codified health records analysis," J. Biomedical Informatics, vol. 162, Art. no. 104761, 2025, doi: 10.1016/j.jbi.2024.104761.

[9] J. Wu, J. Zhu, Y. Qi, J. Chen, M. Xu, F. Menolascina, Y. Jin and V. Grau, "Medical Graph RAG: Evidence-based medical large language model via graph retrieval-augmented generation," in Proc. 63rd Annu. Meeting Assoc. Comput. Linguistics (Vol. 1: Long Papers), Vienna, Austria, 2025, pp. 28443–28467, doi: 10.18653/v1/2025.acl-long.1381.

[10] D. Walke et al., "The importance of graph databases and graph learning for clinical applications," Database, vol. 2023, Art. no. baad045, 2023, doi: 10.1093/database/baad045.

[11] Y. Kim et al., "MDAgents: An adaptive collaboration of LLMs for medical decision-making," in Advances in Neural Information Processing Systems 37, 2024.

[12] S. Schmidgall et al., "AgentClinic: A multimodal benchmark for tool-using clinical AI agents," npj Digital Medicine, vol. 9, Art. no. 499, 2026, doi: 10.1038/s41746-026-02674-7.

[13] W. Shi et al., "EHRAgent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records," in Proc. 2024 Conf. Empirical Methods in Natural Language Processing, 2024, pp. 22315–22339, doi: 10.18653/v1/2024.emnlp-main.1245.

[14] Z. Wang et al., "ColaCare: Enhancing electronic health record modeling through large language model-driven multi-agent collaboration," in Proc. ACM Web Conf. 2025, 2025, pp. 2250–2261, doi: 10.1145/3696410.3714877.

[15] K. Zuo, Y. Jiang, F. Mo and P. Liò, "KG4Diagnosis: A hierarchical multi-agent LLM framework with knowledge graph enhancement for medical diagnosis," in Proc. AAAI Bridge Program on AI for Medicine and Healthcare, PMLR, vol. 281, pp. 195–204, 2025.

[16] Y. Wang et al., "Clinical information extraction applications: A literature review," J. Biomedical Informatics, vol. 77, pp. 34–49, 2018.

[17] E. Yehia et al., "Ontology-based clinical information extraction from physician's free-text notes," J. Biomedical Informatics, vol. 98, Art. no. 103276, 2019.

[18] N. Zhou et al., "Optimal dynamic treatment regime estimation using information extraction from unstructured clinical text," Biometrical Journal, vol. 64, no. 4, pp. 805–817, 2022, doi: 10.1002/bimj.202100077.

[19] B. Njei et al., "Artificial intelligence agents in healthcare research: A scoping review," PLOS ONE, vol. 21, no. 2, Art. no. e0342182, 2026, doi: 10.1371/journal.pone.0342182.

[20] S. Vatsal, H. Dubey and A. Singh, "Agentic AI in Healthcare and Medicine: A Seven-Dimensional Taxonomy for Empirical Evaluation of LLM-Based Agents," IEEE Access, vol. 14, pp. 4840–4863, 2026, doi: 10.1109/ACCESS.2026.3651218.

[21] I. Lopez, A. Swaminathan, K. Vedula, S. Narayanan, F. Nateghi Haredasht, S. P. Ma, A. S. Liang, S. Tate, M. Maddali, R. J. Gallo, N. H. Shah and J. H. Chen, "Clinical entity augmented retrieval for clinical information extraction," npj Digit. Med., vol. 8, Art. no. 45, 2025, doi: 10.1038/s41746-024-01377-1.

[22] H. Luo et al., "HyperGraphRAG: Retrieval-augmented generation via hypergraph-structured knowledge representation," in Advances in Neural Information Processing Systems 38, 2025.

[23] Y. Xie, X. Han, R. Xu, X. Hu, J. Lu and C. Yang, "HypKG: Hypergraph-based knowledge graph contextualization for precision healthcare," in The Semantic Web – ISWC 2025, LNCS, vol. 16140, pp. 328–348, 2026, doi: 10.1007/978-3-032-09527-5_18.

[24] B. Meyer, "Applying 'design by contract'," Computer, vol. 25, no. 10, pp. 40–51, 1992.

APPENDIX A

BOUNDARY-CRITERION EVIDENCE AUDIT

Table A1 reports the retained source locations and concise evidence summaries for the boundary criteria CA4 and NS4 together with the initial independent rater decisions and the final decision used in the manuscript. Evidence summaries paraphrase the retained rater notes; they are not verbatim quotations from the source papers. Where a rater workbook recorded only a broad source location, that label is reproduced rather than assigning a section number retrospectively. Y=Yes, N=No, and * marks a final decision confirmed or revised in a second source audit.

TABLE A1. BOUNDARY-CRITERION EVIDENCE AUDIT FOR CA4 AND NS4

System	CA4 A/B→Final	NS4 A/B→Final	Retained source location / final rationale
MT-clinical BERT	N/N→N	N/N→N	CA4: Sec. 3 Results/Sec. 4—no self-evaluation or recovery loop. NS4: Methodology / Architecture Sections—no symbolic inference, validation, or consistency checking.
CLEAR	N/N→N	N/N→N	CA4: Sec. 4/Discussion—no correction loop. NS4: Sec. 4—symbolic concepts augment context, but no explicit symbolic inference/validation.
ClinicalRAG	N/N→N	N/N→N	CA4: Sec. 3/Evaluation—no autonomous correction loop. NS4: Sec. 2—graph/KB retrieval is reported, not explicit symbolic inference or validation.
ARCH	N/N→N	N/Y→N*	CA4: Sec. 3/Discussion—no autonomous correction. NS4: Sec. 2—graph construction/pruning is offline; second source audit found no runtime symbolic inference/validation.
Medical GraphRAG	N/N→N	Y/N→N	CA4: Sec. 4—no self-evaluation/correction loop. NS4: Sec. 3—structured graph retrieval/verification did not establish explicit rule-based or formal logical inference.
EHRAgent	Y/Y→Y	N/N→N	CA4: Sec. 3 Revision—generated code is revised from execution feedback. NS4: Methodology / Architecture Sections—verification uses code-execution feedback and LLM self-correction, not explicit symbolic inference or validation.
KG4Diagnosis	N/N→N	Y/Y→N*	CA4: Sec. 4—no distinct autonomous correction loop. NS4: Sec. 3.2—entity extraction/relationship reconstruction concerns graph preparation; second audit found no runtime symbolic inference/validation.
MDAgents	N/Y→N*	N/N→N	CA4: Sec. 4—moderator review was not retained as a demonstrated loop that revises an earlier decision. NS4: Methodology / Architecture Sections—validation is natural-language moderator review, not symbolic logic.
ColaCare	N/N→N	N/N→N	CA4: Sec. 4—no demonstrated autonomous correction loop. NS4: Methodology / Architecture Sections—consistency is handled through collaboration/consensus rather than deterministic symbolic validation.
AgentClinic	N/Y→N*	N/N→N	CA4: Sec. 3—doctor-agent behavior did not retain a distinct autonomous correction loop after second audit. NS4: Methodology / Architecture Sections—diagnostic validation relies on LLM reasoning and interaction, not explicit symbolic consistency algorithms.

Note: The complete 80-decision evidence matrix and original rater files are retained by the authors and are available on reasonable request.