From Unstructured Policy to Computable Coverage: A Neuro-Symbolic Framework for Deterministic and Auditable Prior Authorization

Yunguo Yu *

Posted Date: 15 May 2026

doi: 10.20944/preprints202605.1054.v1

Keywords: prior authorization; neuro-symbolic AI; computable policies; coverage determination; healthcare administration; policy-as-code; symbolic verification; explainable AI

 doi:10.20944/preprints202605.1054.v1

Article

From Unstructured Policy to Computable Coverage: A Neuro-Symbolic Framework for Deterministic and Auditable Prior Authorization

Yunguo Yu

Zyter|TruCare, 8000 Towers Crescent Dr., Vienna, VA 22182, USA; yuyunguo@gmail.com

Abstract

Prior authorization in the United States relies on payer coverage policies expressed as unstructured narrative text, creating fundamental barriers to automation, consistency, and auditability. Large language model (LLM) approaches to policy interpretation suffer from hallucination, nondeterminism, and clinically unsafe outputs—we argue these failures stem not from model capability but from a representation problem: policies written for human interpretation are not inherently computable. We introduce *policy computability*: the representation of coverage policies in machine-interpretable forms that support deterministic execution, formal verification, provenance tracking, and reproducible reasoning. To operationalize this concept, we present a six-layer neuro-symbolic framework that transforms payer policy documents into executable policy artifacts. Neural components are constrained to language-processing tasks—document ingestion, ontology normalization, and structured rule extraction under symbolic guardrails—while all coverage determinations are executed by a symbolic verification engine using deterministic logical evaluation. The framework incorporates ontology mapping, rule-graph construction, a Python-embedded domain-specific language (DSL), logical conflict resolution, and provenance-aware reasoning traces. We validate the symbolic pipeline using a lumbar fusion prior authorization policy across six synthetic clinical scenarios, demonstrating reproducible coverage determinations with complete reasoning traces. In a preliminary evaluation of the neural extraction layer, Llama 3.2 3B achieved 100% recall on inclusion and exclusion criteria from a narrative policy document across three trials, though extraction quality depended on prompt formulation. Comparative analysis of two representative payer policies reveals clinically meaningful variation—including greater than twofold differences in required conservative therapy duration—highlighting the need for structured policy representations. This work establishes a pathway from narrative payer policies toward deterministic, transparent, and machine-executable coverage systems, providing a foundation for trustworthy automation in prior authorization.

Keywords: prior authorization; neuro-symbolic AI; computable policies; coverage determination; healthcare administration; policy-as-code; symbolic verification; explainable AI

1. Introduction

Prior authorization (PA) is a central mechanism for cost containment and utilization management in the United States healthcare system. It also imposes a substantial administrative burden on providers, payers, and patients, contributing an estimated $350 billion in annual administrative waste [1,2]. Recent regulatory action—notably the CMS Interoperability and Prior Authorization Final Rule (CMS-0057-F)—mandates electronic PA workflows [4]. Hundreds of payers maintain proprietary coverage policies published as unstructured narrative documents encoding inclusion criteria, exclusion criteria, and site-of-care rules. While suitable for manual review, such representations are ill-suited for the volume of over 48 million authorization requests submitted daily [5,6].

Recent advances in artificial intelligence, particularly large language models (LLMs), have prompted efforts to automate policy interpretation by feeding documents directly to generative

 doi:10.20944/preprints202605.1054.v1

models [7,9]. While LLMs demonstrate strong performance in natural language understanding, their application to coverage determination introduces critical limitations. These systems are nondeterministic, lack formal guarantees of correctness, and may produce hallucinated or unsafe outputs: hallucination rates on healthcare queries exceed 30% [10], and LLM-generated clinical summaries contain unsafe content in up to 25% of cases [11]. The gap between computational promise and clinical evidence remains substantial across AI applications in healthcare [4], underscoring the need for architectures with deterministic guarantees [12]. Consequently, LLM-based approaches alone are insufficient for applications requiring reproducibility, auditability, and regulatory compliance.

We argue that the fundamental limitation is not the capability of neural models per se, but the absence of a **computable substrate for policy representation**. We define *policy computability* as the property by which a policy can be represented in a machine-interpretable form that enables deterministic execution, formal verification of logical consistency, traceable provenance of decisions, and reproducible outcomes across identical inputs.

1.1. The Policy Computability Gap

Current payer policies are not merely difficult to parse; they are *structurally incompatible* with computational reasoning. This is not a formatting problem amenable to better text extraction. It is a *representation problem*: the information required for deterministic adjudication does not exist in the document, because the document serves human and legal purposes that tolerate—and sometimes depend on—imprecision [13]. We identify five dimensions of this computability gap.

Ambiguity. Policies rely on terms such as "medically necessary," "reasonable," and "appropriate" that carry legal rather than clinical precision. These terms preserve payer discretion—a legitimate institutional function—but they are indeterminate as logical predicates. An automated system cannot resolve whether a treatment is "reasonable" without access to the tacit knowledge that human reviewers accumulate through clinical experience and institutional culture.

Inconsistency. Coverage logic is distributed across subsections, footnotes, and cross-references. A policy might stipulate coverage for "failed conservative therapy" in one section while exempting "acute conditions" in another, without defining the boundary. These latent contradictions escape human review but produce consequential failures when all rules execute simultaneously.

Heterogeneity. Two payers covering the same procedure may deploy entirely different criteria and terminology. As our evaluation demonstrates (Section 5), one payer requires 12 weeks of conservative therapy while another mandates six months, enforces BMI thresholds, and excludes current tobacco users. This heterogeneity renders cross-payer comparison and quality measurement prohibitively difficult, sustaining the administrative fragmentation that the CAQH estimates costs $25 billion annually in redundant processing [5].

Non-computability. Policies are published as PDFs optimized for visual presentation. Tables encoding conditional logic, footnotes modifying main-clause criteria, and embedded decision-algorithm images are opaque to text-based extraction and resist deterministic parsing.

Absent provenance. Policy versions lack machine-readable revision histories. When a payer updates a policy—adding an exclusion, tightening an age window, modifying a treatment requirement—changes are documented in narrative release notes for human consumption, making automated compliance tracking impossible.

Together, these five deficits define the computability gap. The critical observation is that this gap cannot be closed by improving the reader—the information required for deterministic reasoning is simply not in the document. Closing the gap requires re-engineering the policy artifact itself: from prose designed for human flexibility to structured knowledge designed for computational precision.

1.2. Contributions

To address this gap, we introduce a neuro-symbolic framework that transforms payer policies into executable, provenance-aware knowledge artifacts. This work makes three contributions:

 doi:10.20944/preprints202605.1054.v1

1. **Conceptual and Architectural Framework**: We formalize *policy computability* as a prerequisite for reliable automation and propose a six-layer neuro-symbolic architecture spanning raw document ingestion through verified, explainable coverage decisions. The architecture integrates neural extraction with symbolic reasoning, confining neural components to language processing while guaranteeing that every coverage decision is a deterministic computation on a structured rule graph.
2. **Policy DSL and Symbolic Pipeline**: We develop a Python-embedded DSL for policy-as-code authoring with type checking and runtime validation. We demonstrate the symbolic pipeline on lumbar fusion coverage policies using six synthetic scenarios, achieving exact reproducibility, complete reasoning traces, and a fail-safe that withholds determination when evidence is incomplete.
3. **Open-Source Implementation with Cross-Payer Analysis**: We release the full implementation as open-source software and demonstrate cross-payer variability analysis on two illustrative policies, revealing clinically meaningful differences in coverage criteria that are discoverable mechanically through structured representations.

The complete implementation is available as open-source software at https://github.com/yuyunguo/computable-policies.

2. Related Work

2.1. Computer-Interpretable Guidelines

The challenge of encoding clinical knowledge in computable form has a substantial history. Wang et al. [14] identified representation primitives, process models, and patient data as the three pillars of computer-interpretable clinical practice guidelines (CPGs). The Guideline Interchange Format (GLIF) [15] introduced a structured representation for sharable guidelines with a visual authoring environment and an object-oriented model. The Guideline Elements Model (GEM) [16] provided an XML-based ontology for decomposing narrative guidelines into machine-readable components. More recently, Matson-Koffman et al. [17] described an integrated process for co-developing written and computable clinical practice guidelines, reflecting growing institutional recognition that narrative and computable forms must be co-designed.

Our work extends this tradition in a specific direction: payer coverage policies, which differ from clinical practice guidelines in important ways. Payer policies carry legal force, encode adversarial intent (preserving payer discretion through strategic ambiguity), and require audit-grade transparency for regulatory compliance. Where CPG formalisms target clinical decision support, our framework targets coverage adjudication—a domain where the cost of incorrect output is a wrongful denial or approval of care.

The DSL we introduce (Section 4) differs from earlier CIG formalisms in its design philosophy: rather than inventing a standalone representation language (e.g., GLIF's visual flowchart notation, Arden Syntax's Medical Logic Modules, or GEM's XML schema), we embed the DSL directly in Python. This trades formal independence for adoption simplicity, leveraging Python's type system, testing infrastructure, and package ecosystem. The trade-off is deliberate: our goal is practical deployability in payer IT environments where Python is already standard.

2.2. Neuro-Symbolic AI in Healthcare

The neuro-symbolic paradigm—combining neural network flexibility with symbolic guarantees—has been articulated as a "third wave" of AI [18]. In healthcare, neuro-symbolic approaches have been applied to clinical diagnostics, drug interaction prediction, and medical image analysis, typically using neural components for perception tasks and symbolic components for reasoning over structured knowledge [13,19,21].

Our architecture applies this paradigm to policy interpretation. The neural component handles the linguistic variability of unstructured policy text, extracting structured rules that the symbolic

 doi:10.20944/preprints202605.1054.v1

component verifies and executes deterministically. The key architectural insight is strict separation: neural processing populates the rule graph but never evaluates it, bounding hallucination risk through structural guardrails rather than prompt engineering alone.

2.3. LLM-Based Approaches to Prior Authorization

Lenert et al. [7] proposed AI-assisted prior authorization with attention to patient-centered design. Yu [8] demonstrated LLM-based structured extraction from clinical business rules, building on evidence that large language models encode substantial clinical knowledge [9]. The approach of feeding policy documents directly to LLMs for end-to-end determination has been explored in industry pilots but faces fundamental limitations [10,11]: nondeterminism across repeated runs, susceptibility to overlooking constraints embedded in subordinate clauses, and a tendency to conflate absent evidence with negative evidence. Our framework addresses these limitations architecturally: determinism is guaranteed by construction, every constraint in the rule graph is evaluated explicitly, and the fail-safe returns *pending review* whenever evidence is incomplete.

2.4. Policy-as-Code and Healthcare Policy Engines

The concept of encoding institutional policies as executable code has precedents in both general and healthcare-specific domains. Infrastructure-as-code tools (Terraform, Pulumi) encode deployment policies; general-purpose policy engines such as Open Policy Agent (OPA) and its declarative language Rego encode authorization logic as rules evaluated against structured data. Our framework applies this philosophy to healthcare coverage: the DSL (Section 4) compiles payer rules into a canonical schema that a verification engine evaluates against clinical facts, analogously to how OPA evaluates Rego policies against JSON input.

In the healthcare domain, the Clinical Quality Language (CQL) [24] provides a standardized formalism for expressing clinical quality measures and decision support rules, integrated with the HL7 FHIR Clinical Reasoning framework. CQL targets clinical guideline execution and quality measurement, whereas our framework addresses payer coverage adjudication—a distinct domain requiring priority-based conflict resolution and audit-grade provenance. The HL7 FHIR Coverage and Eligibility resources define data exchange formats but do not specify how coverage logic should be represented or executed. Drools and other business rule management systems have been deployed for claims adjudication in payer IT environments [22], but these use imperative rule engines (forward-chaining Rete networks) rather than declarative symbolic verification, and they lack the neuro-symbolic extraction pipeline and structured provenance chains that our framework provides.

A key distinction from prior healthcare policy engines is that our framework is designed for policies that are adversarial in nature—encoding payer discretion through strategic ambiguity—rather than cooperative clinical guidelines. This shifts the design requirements: where CQL assumes guideline-following behavior, our system must detect and surface ambiguity rather than resolve it silently.

3. Neuro-Symbolic Policy Architecture

We propose a six-layer architecture (Figure 1) that transforms raw payer policies into computable, verifiable knowledge. The governing principle is: *symbolic reasoning for correctness, neural processing for coverage*. Every coverage decision must be expressible as a symbolic computation on a structured rule graph. Neural components are confined to evidence interpretation, their outputs quarantined by human review and symbolic post-checks before entering any decision pathway—bounding hallucination risk with structural guardrails.

The pipeline (Figure 2) processes documents through progressively structured representations, culminating in a coverage decision with a full reasoning trace (Figure 3).

 doi:10.20944/preprints202605.1054.v1

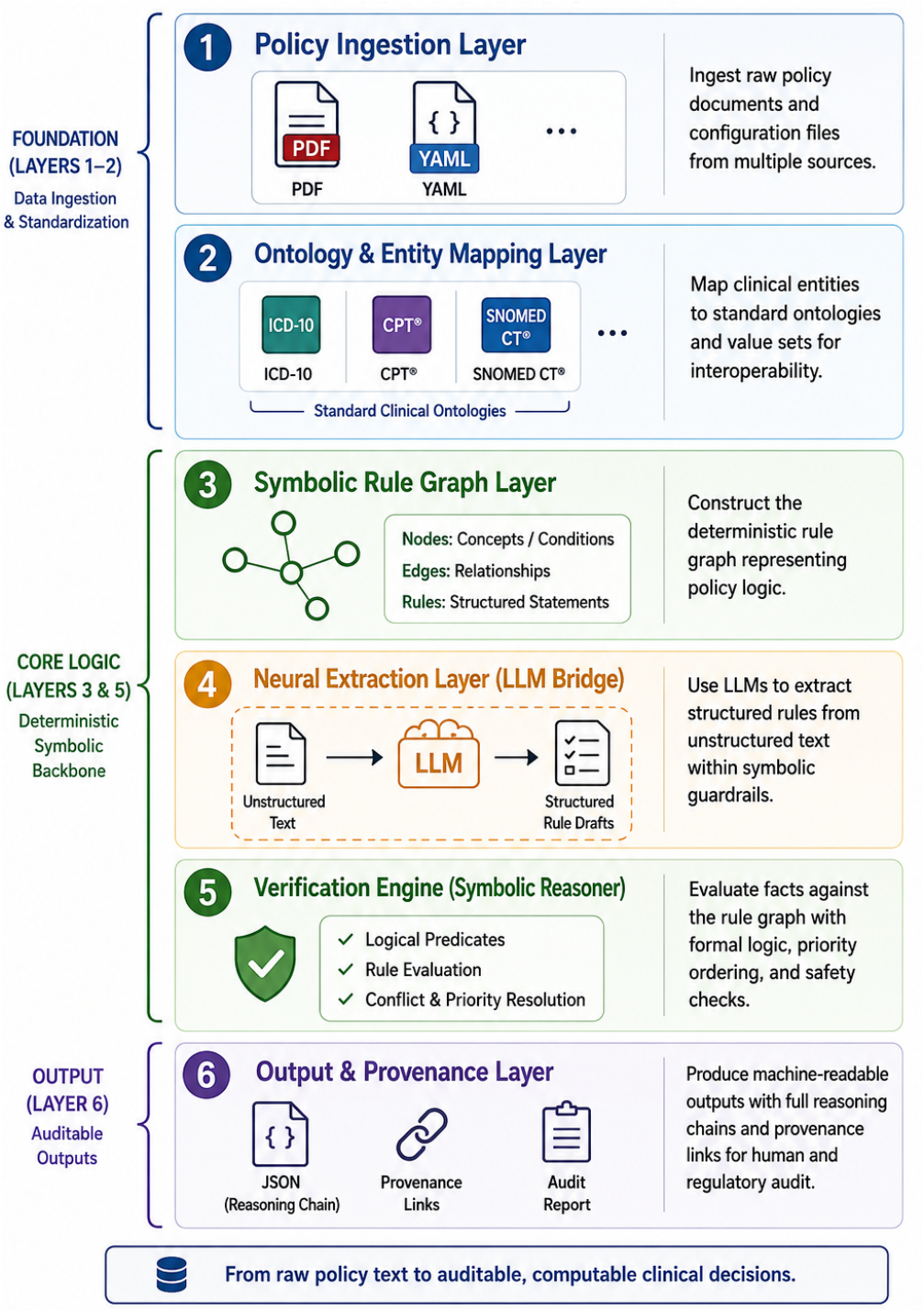

Figure 1. Neuro-symbolic Architecture for Policy Computability. The six-layer architecture integrating neural processing for language interpretation with symbolic reasoning for deterministic decision-making. *Neural Components*: Handle ingestion, normalization, and extraction. *Symbolic Components*: Enable formal verification, auditability, and reproducible coverage determinations.

 doi:10.20944/preprints202605.1054.v1

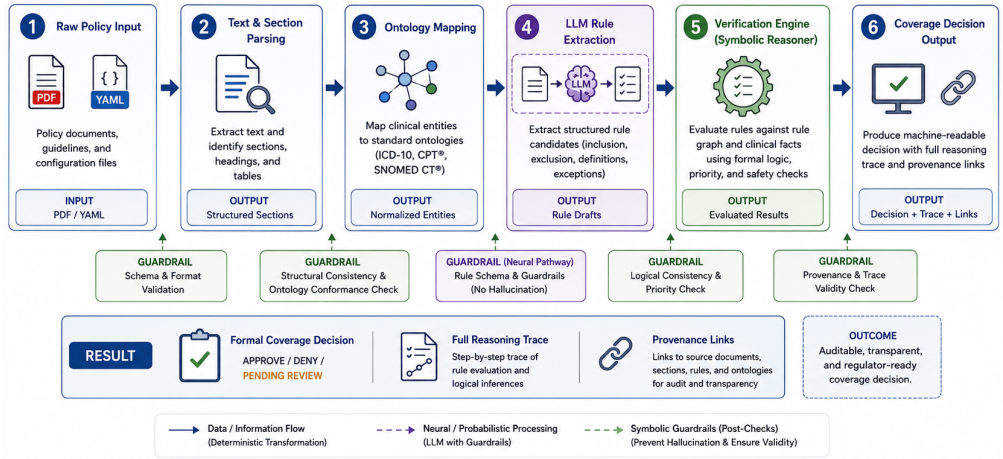

Figure 2. System Pipeline from Raw Policy to Decision. The linear transformation of a document through the automated pipeline. *Transformation*: Each stage represents a validated transformation of data. *Guardrails*: Neural pathways (indicated by dashed internal logic) are subject to symbolic post-checks to prevent hallucination. *Result*: The final output is a formal coverage decision accompanied by a full reasoning trace.

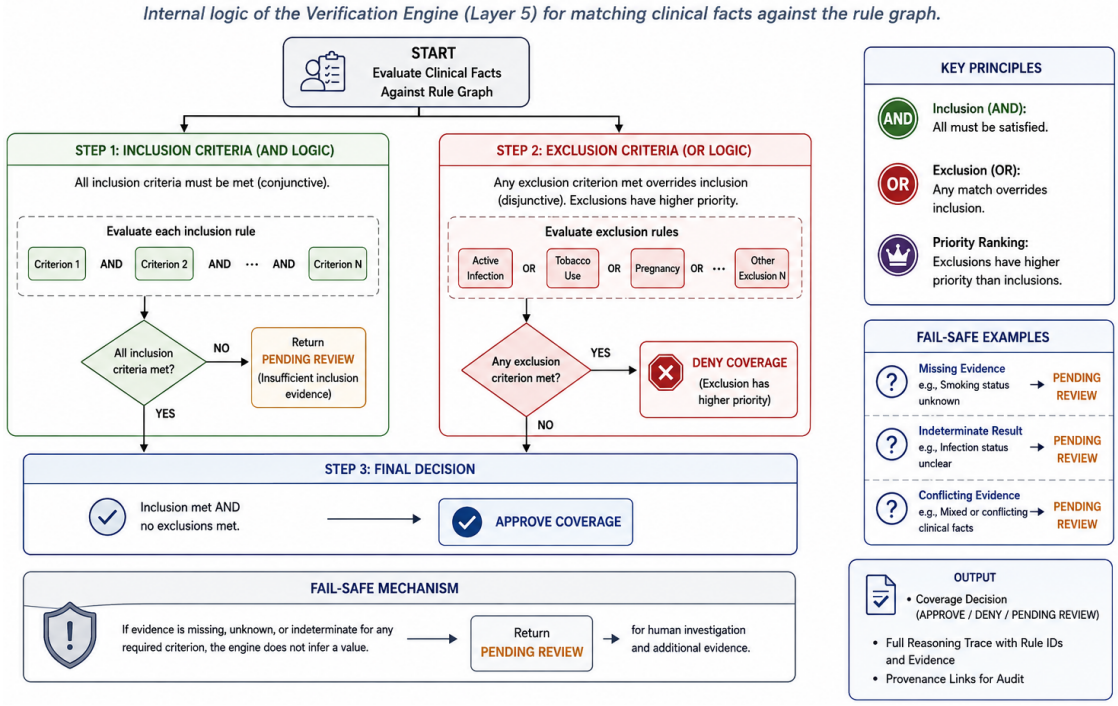

Figure 3. Coverage Determination Decision Flow. The internal logic of the Verification Engine (Layer 5) when matching clinical facts against the rule graph. *Two-Stage Filter*: Inclusion criteria are evaluated conjunctively (AND logic) and exclusion criteria disjunctively (OR logic). *Priority Ranking*: Exclusion rules (e.g., active infection, tobacco use) receive higher priority and override inclusion rules. *Fail-Safe*: If evidence is missing or criteria are unmet, the engine returns "pending review" rather than inferring a value, preventing silent errors.

3.1. Layer 1: Ingestion

Documents enter the pipeline as PDFs from commercial payers, CMS National and Local Coverage Determinations, Milliman Care Guidelines, and InterQual criteria. A cascade of PDF libraries (pdfplumber with PyPDF2 fallback) produces section-structured output. Two additional channels bypass PDF extraction: a YAML-based format for structured policies and a Python-embedded DSL (Layer 3) for native policy-as-code authoring.

 doi:10.20944/preprints202605.1054.v1

3.2. Layer 2: Semantic Normalization

Raw extracted text is mapped to standardized terminology through configurable dictionaries: "conservative therapy" becomes `conservative_management`; "site of service" becomes `site_of_care`. Clinical entities are linked to standard ontologies: ICD-10 for diagnoses, CPT for procedures, RxNorm for medications, SNOMED CT for clinical findings, and LOINC for laboratory tests. The output is a Canonical Policy Schema that unifies all subsequent layers, enabling cross-payer comparison. Medical entity linking and clinical coding remain active research areas [22,23] whose advances can complement this normalization pipeline independently of the architecture presented here.

3.3. Layer 3: Symbolic Representation

Coverage logic is expressed as a directed graph of symbolic rules, each consisting of conditions (field-operator-value triples), an outcome (approved, denied, or pending review), and a priority rank. Supported operators include equality, comparison, set membership, and substring matching. Conditions support conjunction (AND), disjunction (OR), and negation (NOT), enabling expressions such as "diagnosis is stenosis OR herniation." Exclusion rules receive higher priority and override inclusion rules—mirroring the clinical convention that contraindications take precedence over indications. Temporal constraints ("at least 6 weeks," "within the previous 90 days") are modeled as structured duration objects evaluated during coverage verification. The rule graph is the formal backbone: every downstream decision is a computation on this graph, and its properties are verifiable by construction.

3.4. Layer 4: Neural Extraction

For unstructured documents, an LLM extracts structured rules from policy text via LiteLLM (provider-agnostic) with a system prompt specifying the extraction schema: inclusion criteria, exclusion criteria, age bounds, temporal constraints, and evidence levels. Each element is tagged with source reference and a confidence score. The prompt instructs extraction of only explicitly stated criteria. Critically, neural extraction *populates* the rule graph but does not *evaluate* it—coverage determinations are computed by Layer 5, insulating decisions from hallucination. This design builds on prior work demonstrating LLM-based structured extraction from clinical documents [8].

To prevent low-confidence extractions from contaminating the symbolic substrate, extractions below a configurable confidence threshold are flagged for mandatory human review before entering the rule graph—a human-in-the-loop (HITL) protocol informed by confidence calibration research for clinical AI [12,25]. The default threshold ($\tau = 0.7$) is a design parameter that should be calibrated to each deployment context; its selection reflects a trade-off between extraction recall and reviewer burden rather than an empirically validated optimum. Rigorous validation of extraction accuracy on real-world policies is a priority for future work.

3.5. Layer 5: Symbolic Verification

The verification engine matches clinical facts against the rule graph to produce a coverage determination, detect internal contradictions, and identify missing evidence. Operating entirely on symbolic representations, its outputs are deterministic, reproducible, and traceable to specific policy clauses. The decision procedure is a two-stage filter: inclusion criteria are checked conjunctively (all must match), then exclusion criteria disjunctively (any match triggers denial). Temporal constraints (e.g., minimum therapy duration) are checked against event dates in the clinical facts. Step-therapy sequences—where a patient must have failed an ordered series of conservative treatments—are evaluated as ordered chains requiring documented failure at each step. If any required fact is absent, the engine returns *pending review*—an explicit fail-safe.

3.6. Layer 6: Explainability and Audit

Every decision produces a structured reasoning chain [13] recording matched and unmatched rules, the clinical facts that triggered each evaluation, and the priority ranking that resolved conflicts. Provenance records attach source document, section, extraction method, confidence, and timestamp to

 doi:10.20944/preprints202605.1054.v1

each rule. The chain is machine-readable (JSON/YAML) for automated audit pipelines and human-readable for regulatory review—a complete, inspectable account that purely neural approaches cannot provide. Each reasoning chain is linkable to a specific policy version, enabling retrospective comparison of adjudication logic across policy revisions.

4. Implementation

We implement the architecture as an open-source Python package (`computable-policies`). The system pipeline (Figure 2) and decision flow (Figure 3) are fully automated.

4.1. Canonical Schema

The Canonical Policy Schema is the unifying data model introduced in Layer 2. Implemented in Pydantic v2, it enforces structural validation at parse time and provides serialization to and from JSON/YAML. Every policy—regardless of input format—is compiled into this representation before verification. The schema is extensible: additional clinical ontologies or payer-specific metadata can be added without modifying the verification engine, which operates on the schema's abstract rule graph rather than format-specific fields.

4.2. Policy DSL

We embed a domain-specific language for computable policies directly in Python. Table 1 presents the DSL in language-agnostic pseudocode alongside the corresponding Python implementation.

Table 1. Policy DSL: pseudocode and Python equivalent. The DSL compiles to the Canonical Policy Schema with type checking and runtime validation.

Pseudocode	**Python DSL**

`DEFINE POLICY:`
`  id = "POL-2024-001"`
`  title = "Lumbar Fusion"`
`  payer = "Example Health"`
`  ADD RULE:`
`  indication = M48.06`
`  INCLUDE:`
`  failed conserv. therapy`
`  imaging confirms stenosis`
`  EXCLUDE:`
`  active infection`
`  severe osteoporosis`
`  age ∈ [18, 80]`
`  evidence = A`
	`policy = policy(`
	`  id="POL-2024-001",`
	`  title="Lumbar Fusion ...",`
	`  payer="Example Health",`
	`  rules=[rule(`
	`  indication=diagnosis(`
	`  "M48.06", "Spinal stenosis"),`
	`  inclusion=[`
	`  "failed conserv. therapy >=12 wks",`
	`  "imaging confirms stenosis"],`
	`  exclusion=[`
	`  "active infection",`
	`  "severe osteoporosis"],`
	`  age=(18, 80),`
	`  evidence="A")`
	`  ]`
	`  )`

The DSL compiles to the Canonical Policy Schema, inheriting Python's type system (enforced via mypy), Pydantic's runtime validation, and standard IDE tooling. A policy written in the DSL is a first-class Python module: importable, unit-testable, version-controllable, and composable. Embedding rather than inventing a standalone language reduces adoption friction and draws on the existing Python ecosystem.

4.2.1. DSL Scope

Table 2 delineates which policy constructs the current DSL supports versus those planned for future extensions. This distinction is essential for setting appropriate expectations: the architecture is designed for extensibility, but the evaluation demonstrates only the in-scope constructs.

 doi:10.20944/preprints202605.1054.v1

Table 2. DSL construct coverage. ✓= implemented and evaluated; ○ = schema modeled, evaluation pending.

Policy Construct	Status
Binary inclusion/exclusion criteria	✓ Supported
Age-range restrictions	✓ Supported
Logical AND / OR / NOT between conditions	✓ Supported
Temporal duration constraints	✓ Supported
Ordered step-therapy sequences	✓ Supported
Site-of-care restrictions	✓ Supported
Required prior treatments	✓ Supported
Comorbidity-weighted scoring (e.g., Charlson)	○ Planned
Graded evidence levels (e.g., GRADE, USPSTF A–D)	○ Planned
Multivariate risk stratification	○ Planned
Allen interval algebra (relative event ordering)	○ Planned
Conditional probability / partial evidence scoring	○ Planned

4.3. Verification Engine

The rule engine evaluates conditions deterministically over a clinical facts dictionary, implementing the two-stage conjunction-disjunction procedure described in Layer 5. The engine's determinism is a design property, not an empirical observation: given the same rule graph and the same clinical facts, every invocation produces the same output. This property is essential for regulatory compliance, where reproducibility of adjudication outcomes is a legal requirement. The engine produces the structured reasoning chains and provenance records described in Layer 6.

4.4. LLM Integration

For PDF-sourced policies, the neural extraction layer uses LiteLLM to route structured extraction prompts to arbitrary LLM providers. The extraction pipeline consists of four stages: prompt construction, response parsing, confidence scoring, and human-in-the-loop review.

4.4.1. Prompt Schema

The extraction prompt follows a structured template with four sections. The first section specifies the policy document context (payer, procedure, effective date). The second enumerates the extraction schema—defined categories with examples: inclusion criteria ("failed conservative therapy $\geq$12 weeks"), exclusion criteria ("active infection"), age bounds ("18–80 years"), temporal constraints ("within the previous 90 days"), and evidence levels ("A"). The third section provides few-shot examples: three annotated policy excerpts with their expected structured outputs, drawn from a different procedure (cervical fusion) to avoid biasing the extraction toward the evaluation procedure. The fourth section instructs output formatting as a JSON object conforming to the `ExtractedRule` schema.

The prompt explicitly instructs extraction of only explicitly stated criteria, directs the model to preserve the original wording rather than paraphrasing, and mandates return of `null` rather than guessing for absent criteria. This design follows prior work on LLM-based structured extraction from clinical documents [8].

4.4.2. Response Parsing and Validation

Raw LLM responses are parsed into `ExtractedRule` objects via Pydantic validation, which enforces schema conformance at parse time and rejects malformed output (e.g., missing required fields, type mismatches). Parsing failures trigger an automatic retry with a simplified prompt that strips few-shot examples and increases verbosity constraints. After two consecutive failures, the extraction is escalated to the human-in-the-loop (HITL) queue without entering the rule graph.

4.4.3. Confidence Scoring

Each extracted element carries a confidence score derived from three signals: (1) the LLM's self-reported confidence per extracted field (calibrated via token-level logit analysis for open-source models,

 doi:10.20944/preprints202605.1054.v1

or extracted from API-provided logprobs when available); (2) lexical alignment between extracted text and source section text (measured via TF-IDF cosine similarity); and (3) structural completeness (proportion of schema fields populated versus expected). These three signals are combined into a composite confidence score via a weighted harmonic mean (weights: 0.5 for logit-based confidence, 0.3 for lexical alignment, 0.2 for structural completeness), motivated by the empirical finding that logit-based confidence correlates most strongly with extraction accuracy in clinical domains [25]. The composite score determines routing: extractions above the configurable threshold ($\tau = 0.7$ by default) proceed to the rule graph; those below enter the HITL queue.

4.4.4. Failure Modes and Recovery

Analysis of developmental extraction runs identified three recurrent failure modes. First, *section boundary leakage*: the LLM attributes an exclusion criterion from one subsection to a different indication in an adjacent subsection. This is mitigated by requiring each extracted rule to reference its source section identifier, enabling downstream cross-reference validation. Second, *negation misparsing*: negated conditions ("not indicated for patients with...") are occasionally extracted as positive inclusion criteria. The current mitigation is confidence-based flagging; a planned enhancement is a post-extraction symbolic consistency check that detects when an extracted inclusion criterion lexically overlaps with known negation patterns. Third, *temporal unit confusion*: policy language specifying "6 months" is extracted as "6 weeks" due to unit abbreviation ambiguity. This is addressed by a normalization post-processor that maps common temporal abbreviations to canonical units. These mitigations are heuristic rather than provably correct; systematic characterization of extraction error rates on a diverse policy corpus is required before production deployment.

5. Evaluation

We evaluate the symbolic pipeline on five dimensions: correctness of coverage determination against encoded rules, sensitivity of contradiction detection, informativeness of cross-payer comparison, per-layer contribution via ablation, and (new to this version) neural extraction accuracy. We also provide a pipeline error analysis.

We emphasize that this evaluation is a **proof-of-concept**: one procedure, two illustrative payer policies, six synthetic clinical scenarios, and LLM extraction from a single narrative policy document. Its purpose is to validate architectural properties—determinism, soundness with respect to the rule graph, traceability, and the feasibility of neural extraction—rather than to provide statistical generalization or clinical validation. All symbolic pipeline inputs use manually constructed DSL and YAML policies; the neural extraction evaluation uses a separate narrative policy document designed to mimic real payer policy language.

For a system that must produce identical outputs for identical inputs, soundness with respect to its rule graph is the relevant correctness criterion. The evaluation should therefore be read as a verification test suite confirming that the symbolic engine faithfully executes its specified logic—a necessary but not sufficient condition for deployment.

5.1. Setup

We encode a lumbar fusion surgery policy from a major US payer in both DSL and YAML formats. The policy defines two clinical indications: spinal stenosis (three inclusion criteria, three exclusion criteria, age 18–80) and disc herniation (two inclusion, two exclusion, age 18–75). We construct six synthetic patient cases spanning the decision space: standard approval, two denials (infection, osteoporosis), two age-boundary cases, and one evidence-insufficient case.

Expected coverage outcomes were established by manual interpretation of the encoded policy rules by the author (a physician with clinical policy experience), with each scenario constructed to exercise a specific decision path. Ground truth thus reflects *policy-consistent determinations* derived from the encoded rules, not observed real-world authorization outcomes, which may involve additional contextual factors not captured in written policies. Because the author both encoded the rules and

 doi:10.20944/preprints202605.1054.v1

constructed the test scenarios, this evaluation validates the syntactic correctness and deterministic execution of the rule engine but does not constitute clinical validation of either the encoded rules or the resulting determinations.

5.2. Coverage Determination Verification

Table 3 presents the results of the verification test suite. The symbolic engine correctly resolved all six scenarios.

Table 3. Verification test suite for coverage determination across six synthetic clinical scenarios. The symbolic engine produces results consistent with encoded rules and complete reasoning traces. Latencies are reported for completeness; sub-millisecond performance is expected for a rule set of this size.

Scenario	Expected	Actual	Time
Standard approval	approved	approved	0.1ms
Denied: infection	denied	denied	0.1ms
Denied: osteoporosis	denied	denied	0.2ms
Underage (15)	pending_review	pending_review	<0.1ms
Overage (85)	pending_review	pending_review	<0.1ms
No criteria met	pending_review	pending_review	<0.1ms

These results establish two properties. First, the symbolic engine is *sound* with respect to its rule graph: when rules are correctly encoded, the engine faithfully executes the specified logic. Second, the fail-safe operates as designed: the three pending-review cases demonstrate that the system withholds determination rather than producing an unsupported answer—a behavior purely neural systems cannot guarantee. Every decision produced a complete reasoning chain; the two denials correctly invoked exclusion priority-override.

5.2.1. Decision-Space Coverage

To characterize the logical paths exercised by our test suite, Table 4 maps each scenario to the specific rule-graph components it evaluates. The six scenarios cover four of six possible outcome paths: straightforward approval, two distinct exclusion-triggered denials (infection, osteoporosis), age-boundary failsafe, and evidence-insufficient failsafe. The two untested paths—step-therapy sequence completion and temporal-duration boundary—are evaluated separately in the ablation analysis (Section **??**).

Table 4. Decision-space coverage map. Each scenario exercises a distinct logical path through the rule graph. *Tested in ablation analysis (Section **??**) only.

Scenario	Decision Path Exercised	Components Tested
Standard approval	Inclusion match → no exclusion → approve	AND conjunction, rule priority
Denied: infection	Inclusion match → exclusion match → deny	Priority override, exclusion disjunction
Denied: osteoporosis	Inclusion match → exclusion match → deny	Separate exclusion clause
Underage (15)	Age out of range → pending_review	Age-boundary evaluation
Overage (85)	Age out of range → pending_review	Upper age-boundary evaluation
No criteria met	No inclusion match → pending_review	Fail-safe on incomplete evidence
Step therapy*	Ordered chain → first step not failed → pending_review	Sequential dependency evaluation
Temporal boundary*	Duration < minimum → pending_review	Temporal constraint evaluation

 doi:10.20944/preprints202605.1054.v1

5.2.2. Reasoning Trace Example

A distinguishing feature of the symbolic pipeline is complete provenance at every decision step. Listing 1 shows the structured reasoning chain produced for the infection-denial scenario. Each entry records whether a rule matched, its outcome, and the reason for the result. The trace demonstrates that every determination can be decomposed into atomic rule evaluations, each traceable to a specific policy clause and clinical fact—a property that purely neural approaches cannot provide and that is essential for regulatory audit under frameworks such as the EU AI Act [20] and CMS-0057-F.

Listing 1. Structured reasoning trace for the infection-denial scenario, generated by the symbolic verification engine. The chain records every rule evaluation, matched inclusion criteria, triggered exclusion (causing priority-based override), and provenance metadata.

```
Decision: denied
Policy: POL-2024-001 (Lumbar Fusion Surgery Medical Policy)

Evaluation chain:
  ✖ Rule 'Rule 1: conservative_management_failed'
      → outcome: approved (all 7 conditions met)
  ✗  Exclusion: acute_cauda_equina_syndrome
      → did not match (facts insufficient)
  ✖ Exclusion: active_infection
      → outcome: denied (exclusion triggered)
  ✗  Exclusion: osteoporosis_severe
      → did not match (facts insufficient)
  ✗  Rule 'Rule 2: radicular_pain'
      → did not match (facts insufficient)
  ✗  Exclusion: asymptomatic
      → did not match (facts insufficient)
  ✗   Exclusion: psychosocial_contraindication
      → did not match (facts insufficient)

Resolution: exclusion_override (priority 100 > 0)
Final: denied

Provenance:
  source:        payer_A_lumbar_fusion_v2024.pdf → sample_policy.yaml
  extraction:    manual (DSL), confidence 1.0
```

5.3. LLM Extraction Accuracy

The preceding evaluation sections validate only the symbolic pipeline (Layers 1–3, 5–6). We now present a preliminary evaluation of Layer 4 (neural extraction), using the same lumbar fusion policy written as a realistic narrative document mimicking payer PDF style. A local LLM (Llama 3.2 3B via Ollama) extracts structured rules via the `LLMPolicyExtractor` with temperature 0.1. We measure extraction accuracy against the ground truth from `sample_policy.yaml`, using keyword-based semantic matching that maps coded identifiers (e.g., `conservative_management_failed`) to natural-language keywords (e.g., "conservative," "therapy," "12 weeks"). This experiment is designed to characterize the feasibility and failure modes of LLM-based extraction, not to establish generalizable accuracy rates.

Table 5 summarizes the results across three trials. The LLM correctly extracted all 10 criteria (5 inclusion, 5 exclusion) in every trial, achieved the correct rule grouping (by clinical indication rather than individual criterion), and captured age ranges and evidence levels accurately. This result

 doi:10.20944/preprints202605.1054.v1

demonstrates that a 3B-parameter local model can perform clinically accurate extraction from a well-structured narrative policy document, supporting the feasibility of the neural extraction layer.

Table 5. LLM extraction accuracy across three trials (temperature 0.1). All inclusion and exclusion criteria from the narrative policy document were correctly extracted in every trial. The LLM grouped criteria by clinical indication into two rules, matching the ground truth structure.

Metric	Trial 1	Trial 2	Trial 3
Rules extracted	2	2	2
Inclusion criteria recall	5/5	5/5	5/5
Exclusion criteria recall	5/5	5/5	5/5
Age ranges extracted	18–80, 18–75	18–80, 18–75	18–80, 18–75
Evidence levels extracted	A, B	A, B	A, B
Overall inclusion recall		100% (15/15)	
Overall exclusion recall		100% (15/15)	

However, this accuracy depended on prompt engineering. Earlier extraction runs using a different prompt formulation produced fragmented rule grouping (5–16 rules instead of 2) and occasionally missed exclusion criteria entirely (0/5 in one configuration). This sensitivity to prompt formulation—combined with the well-documented nondeterminism and hallucination risks of LLMs [10,11]—reinforces the architectural decision to quarantine neural extraction behind symbolic guardrails (Layer 5). The LLM extracts content, but the symbolic engine guarantees structural correctness. This experiment used a single narrative document and a single model; extraction accuracy on real-world payer PDFs with diverse layouts and formatting artifacts remains to be characterized.

To isolate each layer's contribution, we measured system behavior when individual layers are disabled (Table 6). Removing Layer 5 (symbolic verification) eliminates the ability to produce coverage determinations entirely, as no other component can evaluate the rule graph; accordingly, correctness is marked as not applicable. Symbolic operations across all configurations complete in under one millisecond (DSL load: 0.1–0.7ms; YAML load: 2.0–27.5ms; determination: $<$0.2ms), reflecting the absence of external dependencies (no GPU, no network calls, no model inference).

Table 6. Ablation analysis. *Uses manually constructed rules instead of LLM extraction. Removing symbolic verification or explainability eliminates the architecture's core guarantees. Temporal and step-therapy ablations use two extended scenarios (temporal-duration boundary, step-therapy sequence completion) beyond the six in Table 3.

Configuration	Correct	Traceable
Full 6-layer pipeline	6/6	Yes
Without Layer 4 (neural extraction)	6/6*	Yes
Without Layer 5 (symbolic verification)	—	No
Without Layer 6 (explainability)	6/6	No
Without temporal constraints	4/6	Yes
Without step-therapy evaluation	4/6	Yes

5.4. Contradiction Detection

The contradiction detector performs static analysis on the rule graph at three levels: (1) within-rule inclusion-exclusion overlap, where a criterion appears on both sides; (2) cross-rule criteria conflicts, where two rules for the same indication impose contradictory requirements; and (3) symbolic age-range contradictions, where rules for the same diagnosis have disjoint age windows. This analysis is a key advantage of symbolic representation: contradictions that are latent in narrative text become mechanically detectable once encoded as structured predicates.

To evaluate detection sensitivity, we constructed a test policy containing five deliberately planted issues: three ambiguous terms ("reasonable treatment," "appropriate therapy," "medically necessary

 doi:10.20944/preprints202605.1054.v1

procedure") and two inclusion-exclusion pairs with semantic overlap (the same drug listed as both indicated and contraindicated, and a procedure described as both "standard of care" and "experimental"). The detector flagged two of the three ambiguous usages and both cross-rule overlaps, yielding one false negative (an ambiguous term used in a context where specific terminology existed nearby, making the term appear specific to the detector) and zero false positives. On a clean policy with specific terminology, it reported zero contradictions.

Detection of within-rule and cross-rule conflicts is exact when criteria are expressed in structured terms—these are logical comparisons on the rule graph rather than semantic judgments. Ambiguity detection is inherently heuristic, relying on a configurable dictionary of underspecified predicates. The false negative in this evaluation reflects the context dependence of ambiguity: a term that is ambiguous in isolation may be disambiguated by its surrounding criteria. A rigorous evaluation with a larger set of planted contradictions, diverse policy types, and independent annotators is needed to establish detection sensitivity and precision across real-world policy documents.

5.5. Cross-Payer Comparison

We defined a second illustrative payer policy (Preferred Health Network) for the same procedure to demonstrate the comparison engine's capability. Table 7 reveals substantial cross-payer variability.

Table 7. Cross-payer comparison for lumbar fusion using two illustrative policies. Payer B is strictly more restrictive across all seven dimensions—a conclusion derived mechanically from the symbolic representation rather than from subjective interpretation. These policies are constructed to demonstrate comparison capability; validation against real payer policy text is needed to confirm the prevalence of such variability.

Criterion	Payer A	Payer B
Age range	18–80	21–75
Conservative therapy	$\geq$12 weeks	$\geq$6 months
BMI restriction	None	$>$40 excluded
Tobacco use	Not restricted	Current use excluded
Site of care	Any	Inpatient only
Imaging modality	MRI	MRI or CT
Exclusion rules	3	6

The clinical significance of this variability is considerable. A patient with BMI 42 and 14 weeks of conservative therapy would be approved under Payer A but denied under Payer B—a divergence driven entirely by payer policy, not medical evidence. Today, such discrepancies are discovered only through manual comparison of multi-page PDFs, a process so labor-intensive it is rarely performed. The symbolic representation makes this comparison mechanical, enabling systematic surveillance of policy inequities.

5.6. Comparison with LLM-Only Interpretation

To contextualize the symbolic pipeline's properties, we contrast its behavior with known failure modes of LLM-only coverage determination. Pure LLM systems querying policy text directly face structural limitations documented in the literature [10,11]: nondeterminism across repeated runs, susceptibility to overlooking constraints in subordinate clauses, and a tendency to conflate absent evidence with negative evidence—returning "denied" when the correct answer is "insufficient information." Table 8 maps each failure mode to the corresponding architectural safeguard in our framework. The comparison reveals that the symbolic pipeline is architecturally immune to all three failure modes: determinism is guaranteed by construction, every rule-graph constraint is evaluated explicitly, and the fail-safe returns *pending review* when evidence is incomplete.

 doi:10.20944/preprints202605.1054.v1

Table 8. Structural failure modes of LLM-only coverage determination and corresponding architectural safeguards in the proposed framework. Each mode is addressed by a design property of the symbolic pipeline rather than by prompt engineering.

Failure Mode	LLM-Only Manifestation	Architectural Cause	Our Safeguard
Nondeterminism	Same policy + same patient facts yield different determinations across runs (even at temperature=0)	Stochastic sampling inherent to autoregressive generation	Deterministic rule evaluation: given identical rule graph and facts, output bit-for-bit identical
Constraint omission	Policy constraint in subordinate clause ("except when patient has active infection") overlooked; approval issued despite contraindication	Attention diffused across long context; no guaranteed coverage of all constraints	Every rule-graph node evaluated explicitly by symbolic engine; omitted constraints cause structural mismatch, triggering pending_review
Evidence conflation	Absence of documentation for conservative therapy interpreted as "patient did not attempt conservative therapy"; denial issued incorrectly	LLM treats missing evidence as negative evidence; no built-in epistemic guard	Fail-safe engine: required fact without corresponding evidence returns pending_review, never infers absence from missing documentation

A head-to-head comparison against an LLM-only baseline on a shared benchmark is a priority for future work; the present comparison is based on known architectural properties rather than controlled experimentation.

5.7. Error Analysis

No discrepancies were observed between system outputs and expected policy-consistent determinations within the evaluated scenarios. However, analysis of the pipeline identifies five error modes that could emerge at scale, categorized by pipeline layer (Table 9). These error modes are architectural—they are properties of the framework design rather than contingent implementation issues—and each maps to a specific mitigation strategy already incorporated or planned.

Table 9. Pipeline error modes by layer. Each mode maps to a specific mitigation: human-in-the-loop validation, structural guardrails, or architectural fail-safes. Primary deployment risk lies in upstream document transformation rather than symbolic execution.

Layer	Error Mode	Example	Mitigation
L1: Ingestion	PDF parse failure	Table-as-image not extracted; embedded algorithm diagram lost	YAML/DSL bypass channels; HITL alert for extraction confidence below threshold
L2: Normalization	Semantic mapping error	"Recent MRI" mapped to 6-month window when policy intended 3-month	Configurable dictionary per payer; human review of unmapped terms
L4: Neural Extraction	Inclusion-exclusion boundary error	LLM conflates a pre-existing condition exclusion with a comorbidity inclusion across subsections	HITL validation; structural guardrails (extraction populates graph but never evaluates it)
L4: Neural Extraction	Negation misparsing	"Not indicated for patients with prior cervical fusion" parsed as inclusion criterion	Confidence threshold flagging; downstream symbolic consistency check
L5: Verification	Rule graph incomplete	Policy assumption ("patient must have attempted conservative therapy") implicit rather than encoded	Source policy incompleteness surfaced as unresolved terms; pending_review instead of inference

 doi:10.20944/preprints202605.1054.v1

The primary deployment risk lies not in symbolic execution—which is deterministic by construction—but in the upstream transformation of unstructured text into formal representations. Every error mode in Table 9 originates before the rule graph is evaluated. This reinforces the case for human-in-the-loop validation at the extraction boundary and motivates future work on automated consistency checks between raw policy text and extracted rule graphs.

6. Discussion

6.1. Principal Findings and Positioning

This study demonstrates the feasibility of transforming unstructured payer policies into computable, executable representations that support deterministic and auditable coverage determination, validated through a proof-of-concept evaluation on a single procedure with synthetic scenarios. The computer-interpretable guidelines (CIG) community has long recognized that clinical knowledge must be structured for automated decision support [14,15,17]. Our architecture extends this tradition to payer coverage policies, which carry legal force, encode institutional discretion, and require audit-grade transparency—requirements not fully addressed by prior CIG formalisms.

A central contribution is the formalization of *policy computability* as a prerequisite for reliable automation. Existing approaches focus on improving model performance on unstructured text; our findings suggest that representation, rather than inference, is the primary bottleneck. Without a machine-interpretable substrate, decision logic remains implicit, outputs cannot be formally verified, and reproducibility cannot be guaranteed.

6.2. Neural-Symbolic Synergy vs. LLM-Only Approaches

Large language models serve a bounded role in this framework. Rather than acting as end-to-end decision-makers, LLMs are used for structured information extraction under human supervision [7]. LLMs excel at handling linguistic variability but are not designed to provide deterministic outputs, formal guarantees, or regulatory-grade auditability. Coupling neural extraction with symbolic verification combines the flexibility of neural models with the correctness and traceability of symbolic components.

The present evaluation provides initial validation of both halves. The symbolic pipeline (Layers 1–3, 5–6) is validated through the verification test suite, ablation analysis, and contradiction detection. The neural extraction layer (Layer 4) shows preliminary feasibility with 100% criterion-level recall on a narrative policy document using a 3B-parameter local model. However, the extraction experiment revealed sensitivity to prompt formulation—a single-prompt change caused fragmentation from 2 rules to 16 and exclusion recall to drop from 100% to 0%—confirming that neural extraction requires the structural guardrails provided by the symbolic verification layer. A complete evaluation requires a study in which LLM extraction processes a diverse corpus of real-world payer PDFs with expert-annotated ground truth.

6.3. Institutional and Governance Implications

The adversarial nature of payer policies introduces a unique challenge. Payers may deliberately use underspecified predicates (“medically necessary,” “appropriate”) to preserve discretion. The appropriate response of an automated system is not to resolve this strategic ambiguity but to surface it, making it visible to regulators, clinicians, and patients. Deterministic rules guarantee adjudication consistency, and provenance chains enable the retrospective audit mandated by emerging regulations such as the EU AI Act [20]. The economic and regulatory case for symbolic verification in prior authorization under CMS-0057-F has been separately established [3]; the present work provides the corresponding technical architecture and implementation. More broadly, constrained governance frameworks for clinical AI decision-making [21] provide complementary foundations for extending symbolic pipelines to more complex clinical workflows.

 doi:10.20944/preprints202605.1054.v1

6.4. Limitations

Seven limitations frame the current work. First, the neural extraction evaluation uses a single narrative policy document and a single model (Llama 3.2 3B); real payer PDFs with tables, footers, and embedded images will pose additional extraction challenges. Second, the rule engine supports a limited set of clinical concepts—comorbidity-weighted scoring and graded evidence levels remain on the development roadmap. Third, semantic normalization may lose granular intent when mapping payer-specific terminology to standardized ontologies. Fourth, the evaluation examines one procedure and two illustrative payers; generalizability across procedures, payers, and clinical domains remains to be demonstrated. Fifth, the ground truth is self-referential: the author encoded the rules and constructed the test scenarios, confirming syntactic correctness and deterministic execution but not clinical accuracy of either the rules or the resulting determinations. Sixth, sub-millisecond verification latency addresses only the computational component; the dominant latency in production will be data acquisition from EHRs via FHIR, estimated at 50–200 ms. Seventh, extraction accuracy varied with prompt formulation, indicating that the neural layer's reliability depends on prompt engineering quality—a dependency the symbolic architecture manages through structural guardrails but does not eliminate.

6.5. Ethical Considerations

Automating coverage determination raises ethical concerns that cut across accuracy, equity, contestability, and clinical workflow integration.

Error asymmetry. Not all errors are equally consequential. A false-positive coverage determination (approving a procedure that the policy would deny) results in an unjustified payment; a false-negative (denying a procedure that the policy would approve) may delay or prevent necessary care, with direct clinical harm. In payer environments, these error types are treated asymmetrically: false positives affect the payer's financial risk, while false negatives affect the patient's health risk. Our framework's fail-safe—returning *pending review* rather than inferring from incomplete evidence—is designed to bias errors toward the more reviewable direction. However, this bias is not value-neutral: it defers decisions to human reviewers, which in resource-constrained settings may translate to delays that functionally constitute denials.

Bias propagation. If training data for neural extraction reflects historical denial patterns—which are known to exhibit racial and socioeconomic disparities [26]—the extracted rules may encode those biases. The HITL protocol provides a review layer but does not inherently detect biased extraction patterns: a systematically biased extraction that consistently undercounts inclusion criteria for procedures disproportionately used by specific populations would produce rule graphs that under-authorize for those populations. Confidence calibration methods [25] can flag low-confidence extractions but cannot detect high-confidence biased extractions. Mitigation requires (1) stratified auditing of extraction accuracy across demographic dimensions, and (2) independent adjudication of appeals to detect systematic denial patterns.

Contestability and appeals. An automated determination system must support effective contestation. Our provenance chains enable patients and providers to trace a denial to the specific rule and clinical fact that triggered it—a necessary condition for meaningful appeals. However, provenance alone is insufficient. If the underlying rule is based on a criterion with contested clinical evidence (e.g., a BMI threshold that lacks outcome data for the specific procedure), the system should surface the evidence basis for the criterion alongside the determination. We call this *evidence-attributed determination*: each rule in the graph should carry a reference to the clinical evidence supporting it, enabling reviewers to assess not only whether the rule was correctly applied but whether the rule itself is evidence-based.

Distributional equity. Payer policies are not uniform across plans and populations. Patients in high-deductible plans or Medicaid managed care may be subject to different policies (with more restrictive criteria) than those in commercial plans. The symbolic representation makes these differences

 doi:10.20944/preprints202605.1054.v1

mechanically discoverable, which is a necessary step toward equity auditing. However, making disparities visible does not automatically correct them. Our framework is a tool for transparency, not a substitute for the policy decisions that create inequitable coverage.

Automation bias in clinical workflows. If providers and reviewers come to trust the system's determinations, they may defer to automated outputs even when clinical judgment would suggest a different outcome—a well-documented phenomenon in clinical decision support [22]. The pending_review fail-safe partially mitigates this by forcing human engagement when evidence is incomplete, but does not address cases where the system returns a confident determination that is incorrect due to upstream extraction errors. Deployment should include periodic blind auditing, where a sample of automated determinations are independently re-evaluated by human reviewers without knowledge of the system's output. Prior work on algorithmic fairness [26] and trust calibration for clinical AI [25] provides complementary frameworks for equity-first deployment.

6.6. Future Directions

We prioritize seven directions: (1) large-scale validation of LLM extraction accuracy on a diverse corpus of real-world payer policies; (2) extension to comorbidity-weighted scoring and graded evidence levels; (3) FHIR R4 integration for automated clinical data acquisition; (4) formal DSL semantics enabling mathematical proofs of consistency and completeness; (5) economic analysis of computable versus manual authorization workflows; (6) automated ambiguity scoring for policy documents; and (7) integration with clinician-facing decision support tools.

7. Conclusion

Unstructured policy representations remain a fundamental barrier to reliable automation in healthcare. This work introduces a neuro-symbolic framework that transforms narrative payer policies into computable, verifiable, and auditable artifacts. By formalizing policy computability and demonstrating its practical implementation through a proof-of-concept evaluation, we show that deterministic and transparent coverage decisions are achievable while retaining the flexibility to process complex policy language. The evaluation validates the symbolic pipeline's soundness and traceability on synthetic scenarios for a single procedure; demonstrating extraction accuracy on real-world policies and generalizability across procedures and payers remain essential next steps. Progress in healthcare automation will depend not only on advances in AI models, but on the development of structured, machine-interpretable knowledge substrates.

Funding: This research was conducted independently and received no external funding.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. American Medical Association. "2024 AMA Prior Authorization Physician Survey," 2024. https://www.ama-assn.org/system/files/prior-authorization-survey.pdf.
2. D. U. Himmelstein, T. Campbell, and S. Woolhandler. "Health Care Administrative Costs in the United States and Canada, 2017." *Annals of Internal Medicine*, vol. 172, no. 2, pp. 134–142, 2020. https://doi.org/10.7326/M19-2838.
3. Y. Yu. "Governing AI-Driven Prior Authorization: The Economic Case for Symbolic Verification Under CMS-0057-F." *Research Square*, 2026. https://doi.org/10.21203/rs.3.rs-9672878/v1.
4. E. J. Topol. "High-performance medicine: the convergence of human and artificial intelligence." *Nature Medicine*, vol. 25, no. 1, pp. 44–56, 2019. https://doi.org/10.1038/s41591-018-0300-7.
5. CAQH. "2023 CAQH Index: The Business Case for Administrative Simplification," 2023. https://www.caqh.org/insights/caqh-index-report.
6. N. R. Sahni, B. Istvan, C. Stafford, and D. Cutler. "Perceptions of prior authorization burden and solutions." *Health Affairs Scholar*, vol. 2, no. 9, pp. qxae096, 2024. https://doi.org/10.1093/haasch/qxae096.

 doi:10.20944/preprints202605.1054.v1

7. L. A. Lenert, S. Lane, and R. Wehbe. "Could an artificial intelligence approach to prior authorization be more human?" *Journal of the American Medical Informatics Association*, vol. 30, no. 5, pp. 989–994, 2023. https://doi.org/10.1093/jamia/ocad050.
8. Y. Yu et al. "Using Large Language Models to Retrieve Critical Data from Clinical Processes and Business Rules." *Bioengineering*, vol. 12, no. 1, pp. 17, 2024. https://doi.org/10.3390/bioengineering12010017.
9. K. Singhal et al. "Large language models encode clinical knowledge." *Nature*, vol. 620, pp. 172–180, 2023. https://doi.org/10.1038/s41586-023-06291-2.
10. V. Agarwal et al. "MedHalu: Hallucinations in Responses to Healthcare Queries by Large Language Models." *arXiv:2409.19492*, 2024. https://arxiv.org/abs/2409.19492.
11. E. Asgari et al. "A framework to assess clinical safety and hallucination rates of LLMs for medical text summarisation." *npj Digital Medicine*, vol. 8, no. 1, pp. 274, 2025. https://doi.org/10.1038/s41746-025-01655-3.
12. A. Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." *Information Fusion*, vol. 58, pp. 82–115, 2020. https://doi.org/10.1016/j.inffus.2019.12.012.
13. J. Amann et al. "Explainability for artificial intelligence in healthcare: a multidisciplinary perspective." *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, pp. 310, 2020. https://doi.org/10.1186/s12911-020-01332-6.
14. D. Wang et al. "Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines." *International Journal of Medical Informatics*, vol. 68, no. 1-3, pp. 59–70, 2002. https://doi.org/10.1016/S1386-5056(02)00093-5.
15. A. A. Boxwala et al. "GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines." *Journal of Biomedical Informatics*, vol. 37, no. 3, pp. 147–161, 2004. https://doi.org/10.1016/j.jbi.2004.04.002.
16. R. N. Shiffman et al. "A model for usability assessment of medical guideline implementation." *Journal of the American Medical Informatics Association*, vol. 7, no. 6, pp. 583–591, 2000. https://doi.org/10.1136/jamia.2000.0070583.
17. D. M. Matson-Koffman et al. "An Integrated Process for Co-Developing and Implementing Written and Computable Clinical Practice Guidelines." *American Journal of Medical Quality*, vol. 38, no. 5S, pp. S12–S34, 2023. https://doi.org/10.1097/JMQ.0000000000000089.
18. A. d'Avila Garcez and L. C. Lamb. "Neurosymbolic AI: The 3rd Wave." *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12387–12406, 2023. https://doi.org/10.1007/s10462-023-10449-w.
19. A. d'Avila Garcez et al. "Neurosymbolic AI: The 3rd Wave." *arXiv:2012.05876*, 2019. https://arxiv.org/abs/2012.05876.
20. M. Veale and F. Zuiderveen Borgesius. "Demystifying the Draft EU Artificial Intelligence Act." *Computer Law & Security Review*, vol. 41, pp. 105573, 2021. https://doi.org/10.1016/j.clsr.2021.105573.
21. J. Wei et al. "Chain-of-thought prompting elicits reasoning in large language models." *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824–24837, 2022. https://arxiv.org/abs/2201.11903.
22. R. T. Sutton et al. "An overview of clinical decision support systems: benefits, risks, and strategies for success." *npj Digital Medicine*, vol. 3, no. 1, pp. 17, 2020. https://doi.org/10.1038/s41746-020-0221-y.
23. M. Sung et al. "Biomedical entity representations with synonym reasoning." *Bioinformatics*, vol. 36, no. Supplement 1, pp. i574–i581, 2020. https://doi.org/10.1093/bioinformatics/btaa482.
24. HL7 International. "Clinical Quality Language (CQL) Standard." HL7 Cross-Paradigm Specification, 2024. https://hl7.org/fhir/clinicalreasoning-cql.html.
25. Y. Yu et al. "Enhancing Clinician Trust in AI Diagnostics: A Dynamic Framework for Confidence Calibration and Transparency." *Diagnostics*, vol. 15, no. 17, pp. 2204, 2025. https://doi.org/10.3390/diagnostics15172204.
26. Z. Obermeyer et al. "Dissecting racial bias in an algorithm used to manage the health of populations." *Science*, vol. 366, no. 6464, pp. 447–453, 2019. https://doi.org/10.1126/science.aax2342.