

Typed Neuro-Symbolic Autoformalization of Quantitative Clinical Language: Executable Constraints and Selective Verification

Aleksandra Beliaeva^{1,2,3}

¹Applied AI Institute, Moscow, Russia

²SB AI Lab, Moscow, Russia

³BIMAI-Lab, University of Sharjah, Sharjah, United Arab Emirates

Abstract

Clinical eligibility criteria express numerical comparisons and Boolean relations in prose. We translate them into executable typed formulas using a neural parser, grammar and ontology constraints, and an independent symbolic verifier. The formulas have precise semantics, source-linked predicates, and an auditable record of type, unit, grounding, and consistency checks. On 200 independently double-annotated criteria, a development-selected threshold yields 89.2% logical correctness at 78.5% coverage. At matched 78% coverage, neuro-symbolic, neural, and rule selection achieve 89.7%, 88.5%, and 85.3% correctness, respectively. We evaluate these trade-offs with exact logical equivalence, confidence intervals, and full risk–coverage curves. Structural breakdowns, candidate oracles, augmentation experiments, and controlled edits across two backbones identify Boolean scope and omitted predicates as the main remaining errors. The symbolic interface makes these failures inspectable while separating formal validity from semantic correctness. The study provides an executable approach to quantitative semantic parsing and a controlled testbed for diagnosing its compositional bottlenecks.

1 Introduction

Autoformalization translates natural language into a formal language that can be checked and executed. We study quantitative clinical eligibility criteria. For example, “SBP > 190 or DBP > 95” combines two blood-pressure comparisons. A parser must recover both comparisons and preserve the word “or”.

A simple scope error. Consider the intended formula and a possible parsing error:

$$\begin{aligned}\phi &= (\text{SBP} > 190) \vee (\text{DBP} > 95), \\ \hat{\phi} &= (\text{SBP} > 190) \wedge (\text{DBP} > 95).\end{aligned}$$

For a record with SBP = 200 and DBP = 90 mmHg, ϕ is true and $\hat{\phi}$ is false. Both formulas use valid variables, units, and thresholds. Both can also point to the correct words in the source. Type checking and source-span checks therefore cannot detect this scope error. A comparison with the intended meaning is still necessary.

Our system separates these tasks. A neural parser proposes a Boolean abstract syntax tree (AST). During generation, token masks enforce a grammar and an ontology of supported variables and units. A separate verifier checks the completed AST and records each decision. A selector ranks candidates and can abstain. We test meaning independently by comparing each formula with an adjudicated reference on an exhaustive set of representative records.

The main finding is that Boolean scope and omitted predicates remain the principal bottlenecks. Candidate search, the tested augmentation schemes, and controlled edits on two backbones provide complementary evidence for this diagnosis. Typed formulas and exact evaluation make it possible to distinguish these semantic errors from malformed output, incompatible units, or invalid variables.

Selective prediction provides a practical operating point: the combined selector returns 89.2% logically correct formulas at 78.5% coverage. We compare systems at the same coverage and report full risk–coverage curves, so this accepted-set score can be assessed together with abstention. The symbolic layer also exposes each verification decision and its source evidence, making both accepted outputs and rejections auditable.

Contributions.

- We define a typed Boolean language for quantitative criteria and an exact test of logical equivalence within its supported fragment.
- We combine neural parsing, constrained gen-

eration, symbolic checks, and selective prediction, and state what each component guarantees.

- We evaluate on 200 independently adjudicated criteria and analyze scope errors through structural breakdowns, candidate search, augmentation, and controlled edits across two backbones.

The implementation is available in our [code repository](#). The ontology contains 15 quantitative variables. Our evaluation concerns their formal interpretation; it does not validate clinical decisions.

2 Task and Formal Semantics

Let x be a criterion and ϕ_x its interpretation. A leaf is $p = (v, o, a, u, n, s, e)$: ontology variable v , comparator o , typed value a , canonical unit u , negation flag n , inclusion/exclusion scope s , and source interval e . Internal nodes are AND, OR, and NOT; AND/OR have at least two ordered children and NOT one. Ranges compile to two conjunctive bounds. We retain children in source order. This preserves structure in the representation, but a parser can still omit a required leaf.

Let D_v be the typed domain of v , $\rho(v) \in D_v$ a complete record, and N_u canonical unit normalization. For a numeric leaf,

$$\llbracket p \rrbracket_\rho = I_{n,s}(o(N_u(\rho(v)), a)),$$

where $I_{n,s}$ complements the truth value once if the leaf is negated or has exclusion scope, including when both conditions hold. Categorical equality is defined analogously. Boolean nodes have standard truth-functional denotations. Evidence e and confidence metadata are non-semantic: they are audited but do not change $\llbracket \phi \rrbracket$.

Serialization normal form. The primary JSON Schema permits at most three Boolean edges from the root and at most 12 children per AND/OR node. Variables and units are canonicalized, numeric spellings are normalized, ranges are expanded, and children preserve source order. Semantic AST exact match compares this ordered normal form while excluding evidence and confidence; the artifact hash covers the complete canonical JSON, including evidence. The auxiliary structure-first parser uses a separately reported depth-four, eight-child schema.

Checking meaning exactly. For a single comparison such as $\text{SBP} > 190$, it is enough to test one value below 190, the boundary 190, and one value above it. No comparison changes truth value inside any of these open intervals. We extend this idea to every variable and every threshold in a pair of formulas. For each variable, let T_v be the union of thresholds in the two formulas. The numeric cells are the singleton boundaries $\{t\}$ and open intervals between consecutive thresholds, including the two outer intervals. Finite categorical domains use singleton cells.

Lemma 1 (Threshold-cell completeness). *For formulas over the supported scalar comparisons, canonical units, observed variables, and finite categorical domains, two formulas are logically equivalent iff they agree on the Cartesian product containing every categorical value, every numeric threshold, and one representative from every induced open cell.*

Proof. Every scalar comparison has constant truth value inside each induced cell. Structural induction over AND, OR, and NOT therefore makes each formula constant on every Cartesian-product cell. Agreement on one representative per cell is both necessary and sufficient. $\square$

Thus our reported *logical equivalence* is exact whenever this Cartesian product is exhaustively enumerated; execution agreement is the fraction of its representatives on which denotations agree. Enumeration is exhaustive for all 200 human items and all 207 source-derived items.

3 Neuro-Symbolic Interface

Figure 1 shows the pipeline. Each leaf points to a source span. The verifier records which checks pass, warn, or fail.

3.1 Typed Neural Parser

We use Llama-3.1-8B-Instruct (Grattafiori et al., 2024) with deterministic decoding and a two-epoch LoRA trained on text-to-AST and verifier-derived contrastive records. The output budget is 768 tokens; generation stops after one complete JSON object. The grammar decoder compiles the bounded JSON Schema into token masks. Grammar masks exclude malformed JSON and invalid Boolean arity; ontology masks restrict variable names, value types, categorical domains, and

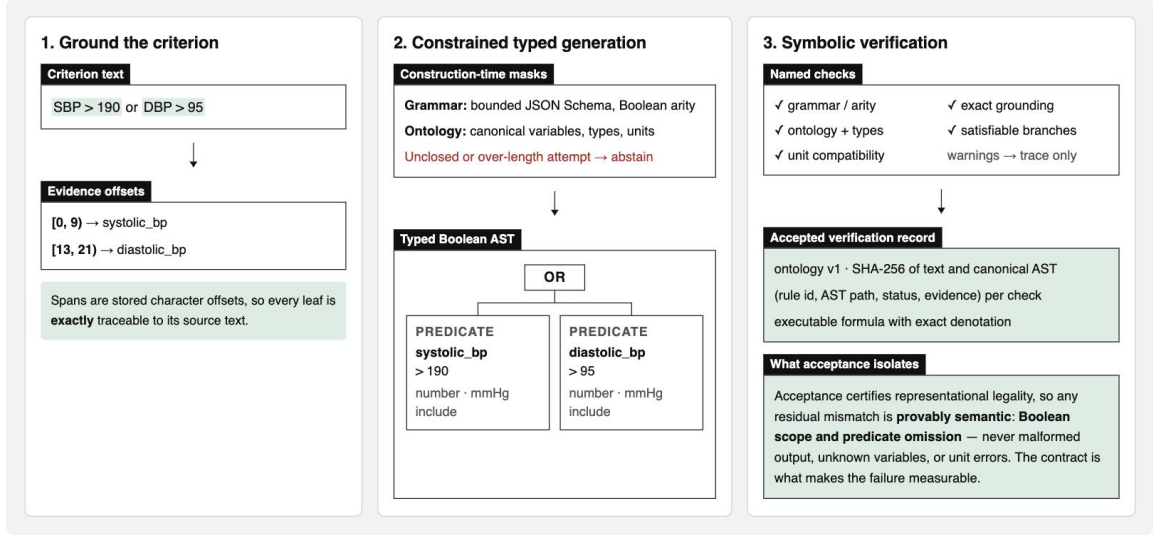


Figure 1: Typed autoformalization and verification. Source spans ground typed AST leaves; named checks expose grammar, ontology, unit, grounding, and consistency decisions. The record explains why a formula passes the implemented checks. It does not prove that the formula expresses the meaning of the text.

variable-compatible units. A length-capped or unclosed attempt abstains, so coverage exposes non-completion.

3.2 Ontology, Verification, and Selection

The declarative ontology contains canonical variables, aliases, value types, comparators, units, and weak plausibility ranges. For $\hat{\phi} = G_{\theta}(x)$, the verifier checks (i) grammar and arity, (ii) canonical variables, comparators, values, and units, (iii) exact evidence offsets and lexical support, (iv) plausibility, and (v) interval consistency in each conjunctive branch. A check is *decisive* exactly when its emitted status is fail; warnings remain in the trace but do not reject. The record stores the ontology version, hashes of x and the complete AST, and tuples (rule id, path, status, evidence).

What acceptance guarantees. An accepted output has valid Boolean arity and ontology-compatible variables, values, and units. Its evidence offsets point to text with lexical support for the leaves, and no implemented interval check finds a contradiction. The record binds these checks to the criterion and AST through hashes and paths. These checks establish formal validity under the implementation. They do not establish that every source predicate is present or that Boolean scope is correct, as the example in Section 1 illustrates. We measure those properties against adjudicated gold.

Compared systems. The *rule parser* has no learned parameters. It uses the released 15-variable alias lexicon, five comparator families, three range templates, two word-order patterns per comparator, and one negated-history pattern. Leaves are aligned to exact source spans; range bounds are joined by AND, and text bridges determine AND/OR composition. The implementation was fixed before final human adjudication. The *indexed parser* first generates a Boolean skeleton whose slots contain character offsets, then fills one typed predicate per slot; contiguous source-ordered slots are required, and its schema permits depth four and eight children. Both stages are trained only on the training split.

The *neural selector* selects from the neural candidate pool using verifier checks. Thus this baseline already uses symbolic verification; it is not an ablation that removes all symbolic components. The *neuro-symbolic selector* ranks one rule candidate, four neural candidates, and two indexed candidates. Its 23 features encode candidate source and index; verifier decision, score, checks, and warnings; evidence rate and mean generation log-probability; agreement with the rule and other neural candidates; and leaf count, depth, and root operator. A standardized ℓ_2 -regularized logistic regression chooses $C \in \{.01, .1, 1, 10\}$ by out-of-fold development selection accuracy, breaking ties toward smaller C . Chia folds are grouped by NCT trial and Criteria2SQL by item. The sigmoid is the score

$r \in [0, 1]$; the threshold is the largest-coverage development point with observed precision at least .85 and a one-sided 95% Clopper–Pearson lower bound of .80. Model, C , and threshold are frozen before test evaluation.

For $g_\tau = \mathbf{1}[r \geq \tau]$, selector coverage is $c(\tau) = N^{-1} \sum_i g_\tau(i)$ and selective logical risk is $R(\tau) = \sum_i g_\tau(i) \mathbf{1}[\phi_i \neq \hat{\phi}_i] / \sum_i g_\tau(i)$. We report the complete attainable risk–coverage curve. Parser coverage means completion of a valid AST; verifier coverage means a deterministic pass; selector coverage means final acceptance. The *Cov* column in the central table always uses the last applicable decision.

Lemma 2 (Sound but incomplete interval rejection). *After NOT elimination, consider an untruncated disjunctive-normal-form clause of canonical, non-negated numeric predicates using $<, \leq, >, \geq, =$. For each variable let L be the maximum asserted lower bound and U the minimum asserted upper bound, retaining endpoint inclusivity. If $L > U$, or $L = U$ with a strict endpoint, the clause is unsatisfiable. A disjunction is rejected as unsatisfiable only when every clause has such a conflict.*

Proof. Any satisfying value must lie in every asserted interval. The stated endpoint conditions make their intersection empty. A disjunction is unsatisfiable only if every disjunct is unsatisfiable. $\square$

The check is incomplete: for example, $x = 0 \wedge x \neq 0$ is contradictory but is not detected by the implemented interval fragment.

4 Evaluation

We derive formulas from supported Chia relations (Kury et al., 2020) and Criteria2SQL conditions (Yu et al., 2020). Chia is grouped by NCT trial; Criteria2SQL official splits are preserved; normalized duplicate texts connect source groups. The release gate verifies zero normalized-text and NCT overlap and no official-test leakage (Appendix A). The resulting source rows number 981/318/442 for train/development/test; the deduplicated neural test has 207 criteria.

A blinded 200-item packet was independently annotated by two general-CS specialists under the closed formal guidelines, followed by third-person adjudication of every serialized disagreement. Annotators saw neither source-derived la-

bels nor model outputs. Semantic AST agreement is .940, Boolean structure agreement .960, and root-operator $\kappa = .950$; exact span agreement is separately analyzed because span boundaries are non-semantic. The packet re-annotates 199 texts from the held-out evaluation pool plus one replacement, so it supplies independent gold rather than a newly sampled corpus.

We compare the rule, neural, indexed, and neuro-symbolic systems defined in Section 3. Metrics are normalized predicate F1, Boolean and full semantic-AST exact match, exact logical equivalence from Lemma 1, execution agreement, satisfiability match, type/unit violations, coverage, and accepted-set quality. Every execution comparison uses the same per-item universe, fixed before system comparison as the union of gold and all candidates in the declared pool; all 200 universes are exhaustively enumerated (at most 243 records). Overall metrics count abstentions as errors; accepted metrics condition on coverage. Exact binomial intervals are labelled *exact*; every bootstrap interval uses 10,000 resamples, clustered by NCT trial for Chia and by item for Criteria2SQL. Of the 200 items, 133 have one leaf, 54 have two, and only 13 have three or more. A range counts as two leaves, so leaf count measures the number of atomic comparisons, not nesting depth alone. This imbalance limits conclusions about complex composition. The appendix reports full metrics and data provenance.

5 Results

5.1 Typed Autoformalization and Selective Reliability

The neural selector has the highest overall logical accuracy: 154/200 (.770), at .980 coverage (Table 1). The rule parser retains 161/200 outputs, of which 135 are correct (.839). At the threshold fixed on development data, the neuro-symbolic selector retains 157/200 outputs. Of these, 140 are correct: .892 accepted accuracy (exact 95% CI [.832, .936]) at .785 coverage. Its accepted execution agreement is .964. It makes no accepted type or unit violations, as do the other systems in the table.

Comparison at the same coverage. The preceding operating points retain different numbers of items. We also compare the frozen rankings at exactly 156/200 items (.780 coverage). Neuro-symbolic, neural, and rule selection then return 140, 138, and 133 correct formulas: .897, .885,

System	Cov.	Overall logical	Accepted logical [exact 95% CI]	Accepted exec.	Viol.
Rule parser	.805	.675	.839 [.772, .892]	.929	.000
Neural selector	.980	.770	.786 [.722, .841]	.915	.000
Indexed parser	.415	.345	.831 [.733, .905]	.940	.000
Neuro-symbolic selector	.785	.700	.892 [.832, .936]	.964	.000

Table 1: Human-gold comparison ($n = 200$). The final threshold is selected on development data; brackets are exact two-sided binomial intervals.

and .853 accepted accuracy. The neuro-symbolic–neural difference is .013, with trial-clustered 95% CI $[-.033, .056]$. This interval includes zero. The .780 point is a descriptive matched-budget comparison, separate from the development-selected .785 threshold.

Across the attainable curves, the neural selector also has a slightly lower normalized area under the risk–coverage curve (AUC): .047 versus .051 for neuro-symbolic selection (lower is better). These values average risk over each system’s available coverage range, ending at .980 and .990 respectively. Figure 2 shows the full curves. Neither the matched-coverage comparison nor AUC establishes a predictive advantage for the combined system. The chosen threshold provides a useful precision–coverage trade-off, while the audit trail supplies a separate benefit.

Variation across training seeds. For the verifier-gated neural selector, independent LoRA runs with seeds 42–44 produce accepted logical accuracy $.797 \pm .017$, execution agreement $.917 \pm .006$, and coverage $.943 \pm .040$ (mean $\pm$ sample SD). Every run has zero accepted type/unit violations. This experiment measures the neural component’s variability; the multi-source selector uses the primary seed. Appendix B.3 gives the per-seed results.

5.2 Scope Errors and Missing Predicates

One expert categorized all 47 logically non-equivalent unconstrained-LoRA outputs using non-exclusive labels. Scope errors occur in 38 cases (80.9%) and missing predicates in 33 (70.2%); comparator direction appears in seven (14.9%) and units in only two (4.3%). This taxonomy describes the unconstrained neural outputs. The structural breakdown below shows that adding verification does not resolve the complex cases either.

Greedy neural accuracy falls from .917 for one leaf to .500 for two and .000 for all 13 formulas with three or more leaves. All three items containing NOT also fail. Neural verifier selection accepts

12 of the 13 larger formulas, and none is correct. Passing verification therefore does not ensure correct composition, even within the closed ontology.

The search diagnostic gives the same warning. Oracle@4 asks whether any of four neural candidates is logically correct. It succeeds on 0/13 larger formulas; adding the rule candidate raises this to only 1/13. The tested composition augmentation does not resolve the failure. Relative to standard verifier selection (.775 overall logical), shallow augmentation yields .735 ($\Delta = -.040$, 95% CI $[-.085, .000]$) and additional four–five-leaf training yields .700 ($\Delta = -.035$ from shallow, $[-.070, -.005]$); all remain 0/13 on formulas with three or more leaves. Because this subset contains only 13 items, these results identify observed failures rather than a precise estimate of performance on complex language. A larger independently annotated set, balanced by leaf count, nesting, coordination, and negation, is needed to test compositional generalization.

5.3 Decoding Changes Scope and Negation Errors

The 100-pair counterfactual suite changes one mathematical-language decision at a time. On both backbones, prompt decoding handles scope edits but fails all ten negation edits; typed+ontology decoding reverses the pattern, recovering all negation edits but none of the ten scope edits (Table 4). Similar aggregate scores therefore hide different kinds of error. The same pattern appears with Falcon3-7B. These are controlled edit families with only ten examples each; they do not demonstrate generalization to unseen natural compositions.

5.4 What the Symbolic Layer Contributes

The clearest benefits are explicit checks and an audit trail. The verifier reports which types, units, source spans, and intervals it checked. A reviewer can inspect a rejected formula or trace an accepted leaf back to the text. This makes failures easier to investigate, even when semantic accuracy does not

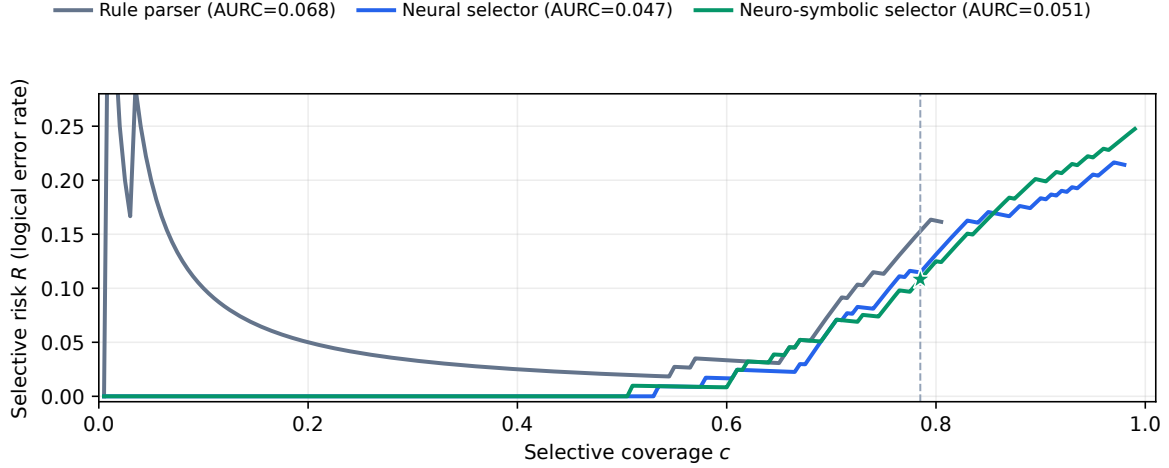


Figure 2: Full logical risk-coverage curves on human gold. Lower risk is better; the star is the development-selected neuro-symbolic operating point. Curves use frozen rankings, and coverage is the fraction of all 200 items retained.

Expert label	<i>n</i>	%	Expert label	<i>n</i>	%
Malformed or abstained	0	0.0	AND/OR/NOT scope	38	80.9
Variable grounding	2	4.3	Missing predicate	33	70.2
Comparator direction	7	14.9	Extra predicate	2	4.3
Numeric value	5	10.6	Evidence grounding	6	12.8
Unit	2	4.3			
Reviewed errors: 47			95 labels		

Gold structure	<i>n</i>	Rule	Neural	Neural+V cov./cond.
1 leaf	133	.865	.917	1.000/.947
2 leaves	54	.389	.500	.944/.549
3+ leaves	13	.077	.000	.923/.000
Contains AND	33	.242	.273	.939/.290
Contains OR	38	.395	.474	.947/.528
Contains NOT	3	.000	.000	1.000/.000

Table 2: Left: non-exclusive expert labels for all 47 neural logical errors. Right: overall exact logical equivalence by overlapping gold-structure groups; “contains” rows may overlap. Neural denotes greedy decoding; Neural+V denotes neural verifier selection and reports coverage/conditional correctness.

improve.

A separate matched-component study finds complementary rule and neural errors. Its evidence-linked hybrid reaches .760 overall logical accuracy, compared with .685 for its neural component and .670 for its rule component: gains of 7.5 points (95% CI [4.0,11.5]) and 9.0 points ([5.0,13.5]). These are different component configurations from Table 1. The result shows that evidence-linked selection can combine complementary candidates. Since it also changes the candidate pool, the gain cannot be attributed to verification alone. The record-integrity audit in the appendix separately tests whether the stored checks detect altered artifacts.

6 Related Work

Autoformalization and typed semantic parsing. Grammar-aware decoders generate programs through abstract syntax (Rabinovich et al., 2017); Neural Symbolic Machines combine neural induction and execution (Liang et al., 2017); and type constraints exclude ill-typed actions (Krishnamurthy et al., 2017). Work on typing variables in

mathematical text (Stathopoulos et al., 2018) motivates our explicit mapping from lexical aliases to typed ontology symbols. We evaluate variable typing together with unit normalization and Boolean meaning, then examine the errors that remain when a formula is well typed.

Constrained decoding and execution guidance. Execution-guided decoding rejects programs that fail during execution (Wang et al., 2018); PICARD incrementally rejects inadmissible SQL tokens (Scholak et al., 2021); and grammar-constrained generation generalizes structured output constraints (Geng et al., 2023). We instantiate JSON-Schema masking with XGrammar (Dong et al., 2025). These methods guarantee membership in an encoded language, not correspondence between text and program. Our controlled edits show how a valid output can still express the wrong Boolean meaning.

Neuro-symbolic verification. Quan et al. (Quan et al., 2024) use a symbolic theorem prover to verify and refine LLM-generated explanations. We instead generate a typed formula under token masks and check the completed output before selection.

Criterion fragment	Human-gold structure	Neural failure
“ALT/AST > 3× upper limit”	ALT > 3 ∨ AST > 3	Shared threshold is not distributed; AST leaf omitted.
“MI within 6 mo and LVEF < 45%”	MI _{6m} ∧ LVEF < 45	First conjunct omitted.
“glucose 200–300 (A1C 7–10)”	Two ranges joined by AND (four leaves)	Parenthetical range and enclosing scope lost.
“bilirubin < 1.5; AST, ALT < 2.5”	Three conjunctive typed leaves	Coordination reduced to the first predicate.

Table 3: Representative adjudicated scope/omission errors. Slash coordination, ellipsis, parenthetical ranges, and shared comparators are characteristic scope phenomena in quantitative prose.

Backbone	Mode	Logical	Local edit	Negation	Scope
Llama-3.1-8B	Prompt	.900	.830	.000	1.000
Llama-3.1-8B	Typed+ontology	.900	.700	1.000	.000
Falcon3-7B	Prompt	.880	.780	.000	.900
Falcon3-7B	Typed+ontology	.870	.550	1.000	.000

Table 4: Frozen counterfactual evaluation ($n = 100$ per row; ten edits per family). Local edit requires that exactly the intended AST path change.

Our checks concern formal validity and source links. They do not prove that the text entails the formula. We evaluate this remaining semantic question against human gold.

Clinical quantitative language and selective prediction. Chia annotates trial criteria (Kury et al., 2020), Criteria2SQL maps criteria to queries (Yu et al., 2020), Valx structures numerical comparisons (Hao et al., 2016), and clinical NLP studies relations among numerical expressions (Dasgupta et al., 2020). We jointly evaluate typing, Boolean structure, exact execution, and evidence grounding. Coverage-conditioned reporting follows selective prediction (Geifman and El-Yaniv, 2019); we choose the operating point on development data and do not infer calibration from a hand-built score (Guo et al., 2017).

7 Conclusion

We present a typed neuro-symbolic pipeline that turns quantitative eligibility language into executable Boolean formulas. Its main contribution is a clear separation between neural interpretation, formal checks, and exact evaluation of meaning. Source-linked verification records make the output auditable, and selective prediction yields 89.2% logical correctness at 78.5% coverage.

The combined evidence from structural analysis, candidate search, augmentation, and controlled edits identifies Boolean scope and predicate omission as the main remaining bottlenecks. This diagnosis directs future work toward recovering complete predicate sets and their composition. Clinical eligi-

bility language thus provides a controlled testbed for both executable semantic parsing and the study of mathematical-language understanding.

8 Limitations

Benchmark size and composition. The ontology contains 15 variables and 45 aliases. Of the 200 human-gold formulas, 187 have at most two leaves and only 13 have three or more; only three contain NOT. Aggregate results primarily describe shallow criteria. The neural system returns no correct formula on the 13 larger items; this small subset limits precise estimates across types of composition. The human packet re-annotates 199 held-out texts plus one replacement, rather than introducing an independently sampled corpus.

Verification and predictive accuracy. The verifier checks implemented formal constraints, lexical support, and record integrity. It can accept missing predicates or incorrect scope. Its interval reasoning is incomplete. Moreover, the neural selector already uses verifier checks, so comparing it with the combined selector does not isolate the effect of adding symbolic verification. The matched-coverage confidence interval includes zero, and the neural selector has better overall accuracy and AURC. The development precision criterion is not a guarantee of test precision.

Annotation and robustness. Two general-CS specialists annotated formal interpretations, followed by third-person adjudication; this is not clinical validation. Evidence boundaries vary between minimal predicates and enclosing clauses, although aligned spans overlap. One expert provided the error taxonomy. The main comparison uses training seed 42; the three-seed experiment measures the neural component, not the full multi-source selector. The second backbone and small controlled edit families provide limited replication, not broad model or language coverage.

Clinical scope. We use public eligibility text and no private patient records. The semantics assume complete records and the supported ontology. Clinical judgment, missing patient information, and criteria outside this fragment are beyond this study. The outputs are research artifacts, not clinical decisions.

References

- Tirthankar Dasgupta, Ishani Mondal, Abir Naskar, and Lipika Dey. 2020. [Exploring the role of semantic relationships in numerical expression recognition](#). In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 243–248. Association for Computational Linguistics.
- Yixin Dong, Charlie F. Ruan, Yaxing Cai, Ziyi Xu, Yilong Zhao, Ruihang Lai, and Tianqi Chen. 2025. [XGrammar: Flexible and efficient structured generation engine for large language models](#). In *Proceedings of Machine Learning and Systems*, volume 7.
- Yonatan Geifman and Ran El-Yaniv. 2019. [SelectiveNet: A deep neural network with an integrated reject option](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2151–2159. PMLR.
- Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. [Grammar-constrained decoding for structured NLP tasks without finetuning](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952. Association for Computational Linguistics.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. [The Llama 3 herd of models](#). *arXiv preprint arXiv:2407.21783*.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. [On calibration of modern neural networks](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.
- Tianyong Hao, Hongfang Liu, and Chunhua Weng. 2016. [Valx: A system for extracting and structuring numeric lab test comparison statements from text](#). *Methods of Information in Medicine*, 55(3):266–275.
- Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. 2017. [Neural semantic parsing with type constraints for semi-structured tables](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1526. Association for Computational Linguistics.
- Fabricio Kury, Alex Butler, Chi Yuan, Li-Heng Fu, Yingcheng Sun, Hao Liu, Ida Sim, Simona Carini, and Chunhua Weng. 2020. [Chia, a large annotated corpus of clinical trial eligibility criteria](#). *Scientific Data*, 7(1):281.
- Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. 2017. [Neural symbolic machines: Learning semantic parsers on Freebase with weak supervision](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–33. Association for Computational Linguistics.
- Xin Quan, Marco Valentino, Louise A. Dennis, and Andre Freitas. 2024. [Verification and refinement of natural language explanations through LLM-symbolic theorem proving](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2933–2958. Association for Computational Linguistics.
- Maxim Rabinovich, Mitchell Stern, and Dan Klein. 2017. [Abstract syntax networks for code generation and semantic parsing](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1139–1149. Association for Computational Linguistics.
- Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. [PICARD: Parsing incrementally for constrained auto-regressive decoding from language models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901. Association for Computational Linguistics.
- Yiannos Stathopoulos, Simon Baker, Marek Rei, and Simone Teufel. 2018. [Variable typing: Assigning meaning to variables in mathematical text](#). In *Proceedings of NAACL-HLT 2018, Volume 1 (Long Papers)*, pages 303–312. Association for Computational Linguistics.
- Chenglong Wang, Kedar Tatwawadi, Marc Brockschmidt, Po-Sen Huang, Yi Mao, Oleksandr Polozov, and Rishabh Singh. 2018. [Robust text-to-SQL generation with execution-guided decoding](#). *arXiv preprint arXiv:1807.03100*.
- Xiaojing Yu, Tianlong Chen, Zhengjie Yu, Huiyu Li, Yang Yang, Xiaoqian Jiang, and Anxiao Jiang. 2020. [Dataset and enhanced model for eligibility criteria-to-SQL semantic parsing](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5829–5837, Marseille, France. European Language Resources Association.

A Evaluation Provenance and Audit Trail

This appendix follows one evaluation item from its source corpus to an adjudicated formula, a finite denotational test, and an auditable verification record. It first fixes data identity and counting conventions, then reports the label and artifact audits that support the main results.

Measure on 200 shared texts	Agreement
Predicate F1	.902
Boolean structure exact	.800
Semantic AST exact	.775
Logical equivalence	.785 [.730, .840]
Execution agreement	.922

Table 5: Source-derived labels compared with adjudicated human gold.

A.1 From Source Corpora to Human Gold

Chia is grouped by NCT trial, Criteria2SQL preserves its official assignments, and normalized duplicate texts connect source groups. The release gate requires zero normalized-text overlap, zero Chia NCT overlap, and no official test or development leakage. One contradictory duplicate is excluded in full.

The human benchmark contains 200 formulas with two independent annotations and adjudication. Source-derived labels are deterministic transformations used only for the secondary factorial; counterfactual labels are programmatically generated controlled edits. Table 5 quantifies how the source-derived formulas agree with human gold on the same packet texts, rather than treating the two label sources as interchangeable.

A.2 Counting and Denotational Evaluation

Ranges are expanded into two conjunctive leaves before leaf counts are computed. The size strata 1/2/3+ are disjoint and total 200. “Contains AND”, “Contains OR”, and “Contains NOT” inspect the complete compiled AST and overlap: a nested formula can contribute to several operator rows, and NOT may wrap a one-leaf formula. Operator counts therefore need not equal the number of non-atomic formulas.

For each item, one evaluation universe is fixed from the gold AST and every candidate in the rule, neural- $k = 4$, and indexed- $k = 2$ pools. Threshold cells are enumerated exhaustively; the largest Cartesian product contains 243 records. Logical equivalence is unchanged by this shared-universe rescore. All bootstrap intervals use 10,000 resamples with NCT-level Chia clusters and item-level Criteria2SQL clusters; exact binomial intervals are labelled separately. Table 6 gives the complete central metric vector.

A.3 Evidence and Verification-Record Integrity

Every record stores the ontology version, SHA-256 hashes of the criterion and canonical AST, and named checks with AST paths and offsets. Across 279 aligned annotator span pairs, all overlap: 59 are exact and 220 nested (character IoU .526; token F1 .659). Thus boundary disagreement reflects minimal-predicate versus enclosing-clause annotation rather than disjoint textual evidence.

The artifact audit injects 2,035 deterministic mutations spanning 11 implemented fault classes. Re-audit detects and localizes every injected fault, while 185 unmodified accepted records produce no false rejection. This experiment tests the verification guarantee in Section 3—hashes, paths, offsets, types, and encoded consistency checks—separately from semantic entailment.

A.4 Data Statement

Chia consists of public trial eligibility criteria, contains no patient records, and is distributed under CC BY 4.0 (Figshare DOI 10.6084/m9.figshare.11855817.v2). The Criteria2SQL repository carries Apache 2.0 for its code, but its README

does not state a separate license for corpus text. The code repository provides download and transformation scripts. Users obtain the underlying corpus texts from their original repositories; downloaded texts and derived annotations are kept outside the public code repository. The artifacts are intended for research on formalization, not clinical decision making.

B System Reproduction and Robustness

The second part moves from evaluation artifacts to the systems that produce predictions. We specify the two auxiliary parsers and selector first, then give training details, independent-seed stability, and the mechanism factorial.

B.1 Rule, Indexed, and Selection Components

The rule parser contains five comparator families, three range templates, ten comparator-specific word-order patterns, and one negated-history pattern. It has no fitted parameters and was fixed before final adjudication. The indexed parser generates a depth-four Boolean skeleton with source character offsets, validates contiguous ordered slots, then fills one typed leaf per slot. Its final evidence spans are overwritten by the deterministic skeleton offsets.

The selector has 23 features: three source indicators; candidate index; verifier acceptance and reliability; evidence rate; mean log-probability; full-pool and available-pool support; agreement with the rule and another neural candidate; leaf count; depth; four root indicators; check count; pass and warning fractions; and mean/minimum check confidence. We standardize these features and fit ℓ_2 -logistic regression in five development folds that keep each Chia NCT trial intact. The selected $C = .1$, sigmoid score, and threshold .568 are frozen before test scoring.

B.2 Training and Inference Configuration

Table 7 lists the training and inference settings.

B.3 Independent-Seed Stability

We independently train the same LoRA configuration with seeds 42, 43, and 44. Candidate generation uses seed 2026 in every run, isolating variability from parameter training. Table 8 evaluates the frozen verifier-gated neural selector on the same 200 human-gold items. Accepted logical and execution quality remain stable; most between-seed variation appears in coverage and consequently in overall logical accuracy.

B.4 Mechanism Factorial

The final diagnostic separates adaptation, token masks, and post-generation verification on 207 source-derived items. Typed decoding guarantees construction-time type safety, while adaptation supplies the semantic improvement over the base model. These cached generations use the same exhaustive threshold-cell evaluator as the human-gold experiment.

The appendices document the data, annotations, evaluation, and implementation needed to interpret the reported results.

System	Cov.	Pred. F1	Bool.	AST	Overall log.	Accepted log.	Sat.
Rule parser	.805	.756	.655	.650	.675	.839	.805
Neural selection	.980	.866	.830	.750	.770	.786	.980
Selective unified	.785	.732	.720	.690	.700	.892	.785

Table 6: Complete central metrics on adjudicated human gold.

Setting	Value
Primary model	Llama-3.1-8B-Instruct
Training data	3,698 records: 734 text-to-AST and 2,964 verifier-derived contrastive records
Development data	1,445 records: 254 text-to-AST and 1,191 contrastive records
Optimization	2 epochs; batch 2; gradient accumulation 4; learning rate 2×10^{-4} ; linear schedule; no warmup; weight decay 0; BF16
LoRA	Rank 16; $\alpha = 32$; dropout .05; q/k/v/o_proj; 13,631,488 trainable parameters
Lengths	1,536 training tokens; 768 generated tokens; stop after one complete JSON object
Typed decoding	XGrammar 0.1.29; strict JSON-Schema masks; deterministic decoding
Ontology	Version 1.0.0; 15 variables; 45 aliases
Software	Transformers 4.47.0; PEFT 0.18.1; PyTorch 2.5.1

Table 7: Training and inference configuration.

Training seed	Cov.	Overall logical	Accepted logical	Accepted exec.	Viol.
42	.980	.770	.786	.914	.000
43	.950	.775	.816	.924	.000
44	.900	.710	.789	.914	.000
Mean $\pm$ SD	.943 $\pm$.040	.752 $\pm$.036	.797 $\pm$.017	.917 $\pm$.006	.000

Table 8: Independent LoRA training seeds on human gold ($n = 200$ per seed). The final row reports mean $\pm$ sample standard deviation.

System	Coverage	Logical equivalence	AST	Execution	Viol.
Base unconstrained	.000	.000	.000	.000	.000
Base unconstrained + verifier	.000	.000	.000	.000	.000
Base typed	1.000	.507	.464	.795	.000
Base typed + verifier	.556	.765	.748	.900	.000
LoRA unconstrained	1.000	.802	.778	.919	.019
LoRA unconstrained + verifier	.918	.847	.832	.942	.000
LoRA typed	.995	.738	.718	.908	.000
LoRA typed + verifier	.923	.775	.759	.916	.000

Table 9: Source-derived factorial ($n = 207$); metrics condition on coverage.