From Argumentation to Labeled Logic Program for LLM Verification

Boris A. Galitsky *

Posted Date: 21 January 2026

doi: 10.20944/preprints202601.1549.v1

Keywords: large language models; hallucination detection; labeled logic programs; neuro-symbolic reasoning; argumentation; abductive inference; discourse-aware verification

 doi:10.20944/preprints202601.1549.v1

Article

From Argumentation to Labeled Logic Program for LLM Verification

Boris A. Galitsky

Knowledge Trail Inc, San Jose, CA; bgalitsky@hotmail.com

Abstract

Large language models (LLMs) often generate fluent but incorrect or unsupported statements, commonly referred to as hallucinations. We propose a hallucination detection framework based on a Labeled Logic Program (LLP) architecture that integrates multiple reasoning paradigms, including logic programming, argumentation, probabilistic inference, and abductive explanation. By enriching symbolic rules with semantic, epistemic, and contextual labels and applying discourse-aware weighting, the system prioritizes nucleus claims over peripheral statements during verification. Experiments on three benchmark datasets and a challenging clinical narrative dataset show that LLP consistently outperforms classical symbolic validators, achieving the highest detection accuracy when combined with discourse modeling. A human evaluation further demonstrates that logic-assisted explanations improve both hallucination detection accuracy and user trust. The results suggest that labeled symbolic reasoning with discourse awareness provides a robust and interpretable approach to LLM verification in safety-critical domains.

Keywords: large language models; hallucination detection; labeled logic programs; neuro-symbolic reasoning; argumentation; abductive inference; discourse-aware verification

Introduction

Large language models (LLMs) have achieved impressive results across a wide range of natural language processing tasks, generating fluent and informed text. Yet integrating them into domains that demand structured, context-sensitive reasoning remains difficult. LLMs often rely on associative rather than strategic reasoning, which limits their ability to perform multi-step decision-making or revise conclusions as new information emerges (Kalai 2025).

Another challenge lies in interpretability. Unlike human experts who reason through explicit, traceable arguments, LLMs operate as opaque statistical systems, making it hard to justify their conclusions or detect reasoning errors. This opacity fosters reasoning hallucinations—outputs that sound plausible but contradict facts or logic. Without explicit mechanisms for defeasibility or conflict resolution, such inconsistencies undermine reliability in high-stakes applications.

To address these issues, LLMs can be coupled with external reasoning and verification layers that enforce logical consistency and explain conclusions. A promising strategy is pairing an LLM with a symbolic reasoning engine—such as a Prolog-style rule base, constraint solver, or medical ontology (Galitsky 2025). The LLM proposes candidate answers, while the reasoning module tests them against formal rules, flagging contradictions or unsupported claims. Building on this principle, we present ValidLogic4LLM, a neuro-symbolic verification framework that externalizes and evaluates LLM reasoning through various forms of reasoning:

1. Logic programming,
2. Probabilistic logic programming,
3. Argumentation,
4. Abductive explanation.

 doi:10.20944/preprints202601.1549.v1

In the proposed framework, ValidLogic4LLM, we use LLM to build respective logic program components for user request and background knowledge, execute this logic program and prompt LLM to compare its run with LLM own result.

The key contributions of this demo paper are as follows:

1. Neuro-symbolic integration: We introduce a framework that integrates logical reasoning components with LLMs, enabling the model to reason over decisions structured according to discourse relations.
2. Hallucination challenge dataset: We develop a benchmark dataset intentionally designed to induce hallucinations in LLMs through adversarial and ambiguous prompts, serving as a testbed for evaluating reasoning robustness.
3. Logic-based hallucination detection: We demonstrate that the logical reasoning module can successfully detect and explain hallucinations by cross-checking LLM outputs against discourse-structured rules and ontology-derived facts.

Figure 1 compares four levels of reasoning used by language models, showing how they evolve from simple answers to structured, self-correcting discourse-based reasoning. At the top, the direct answer model provides an immediate response without explanation. It may be correct or incorrect, but there is no visibility into why the model chose that answer. Because no reasoning steps are revealed, errors cannot be traced or corrected.

The next level, chain-of-thought reasoning, adds a sequence of intermediate steps that make the process more interpretable. However, these steps remain unverified. The reasoning might sound plausible while still being factually wrong, since the model does not test or challenge its own statements.

The argumentative model introduces a more structured approach. Instead of producing one line of reasoning, it generates multiple arguments, distinguishing between supporting and attacking ones. This enables a form of contestability: each conclusion is backed by explicit evidence and can be challenged by counterarguments. Still, this stage only formalizes reasoning—it does not validate or improve it. The model outputs argument structures but lacks a mechanism to revise its own conclusions.

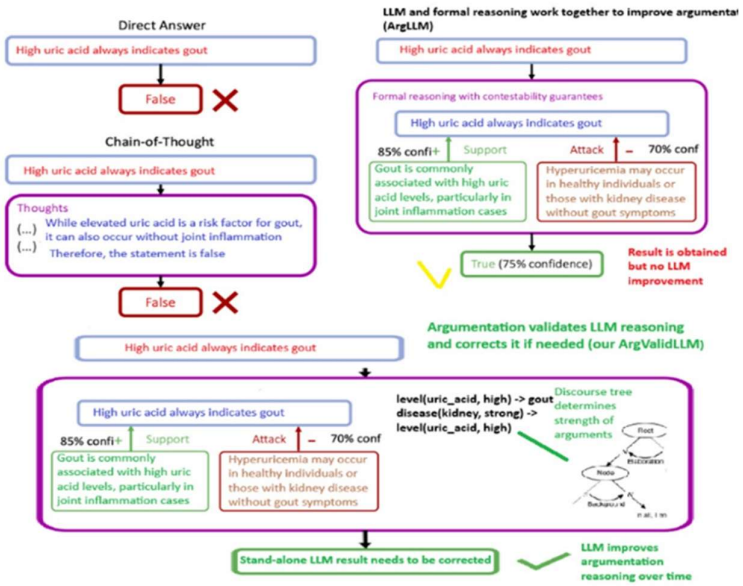

Figure 1. An illustration of our idea of ValidLogic4LLM's reasoning components validating LLM results.

 doi:10.20944/preprints202601.1549.v1

At the bottom, the discourse-based validation model ValidLogic4LLM, the full multi-logic ensemble, represents the most advanced reasoning form. It integrates these four logical verifiers with discourse structure analysis, evaluating how strongly each inference contributes to the overall reasoning. By analyzing rhetorical relations such as elaboration, justification, and background, it weighs the importance of each inference and determines which should dominate the final conclusion.

This allows the system to detect when the original model's answer is inconsistent with the discourse-level balance of evidence and to automatically correct it. Over time, this validation loop enables the model to refine its reasoning and become more consistent, explainable, and logically grounded. The architecture thus moves beyond producing or scoring arguments—it combines multiple logical paradigms within a discourse-aware verification layer to ensure that reasoning outcomes are coherent, probabilistically justified, and defensible.

To cover an arbitrary form of reasoning, we employ labeled deductive reasoning approach and implement it via what we call a labeled logic program.

Labeled Logic Programs for Logic-Agnostic Verification of LLM Reasoning

Verifying LLM outputs is difficult because the type of reasoning required is often unknown in advance. A generated explanation may rely on temporal ordering, causal relations, modal distinctions (what is known or believed), defeasible argumentation (exceptions and priorities), or probabilistic uncertainty. Traditional verification approaches usually assume a fixed logical formalism, which limits their ability to handle such heterogeneous reasoning patterns.

To address this, ValidLogic4LLM adopts the idea of Labeled Deductive Systems (LDS), introduced by Gabbay and Woods (1993), where inference is performed over labeled formulas of the form "$t : A$". Here, A is an object-level statement and t is a structured label that stores meta-information such as time, source reliability, modality, argumentative role, or uncertainty. Different logics are obtained not by changing the object language, but by changing how labels are interpreted, combined, and propagated. This makes LDS well suited for logic-agnostic verification.

ValidLogic4LLM specializes this idea to Labeled Logic Programs (LLPs). In an LLP, both facts and rules are annotated with labels. For example, a rule like "complication(X):- infection(X), immunosuppressed(X)" can be associated with a label that specifies its evidential status, typical confidence, or applicable context. When the rule is applied, the labels of the premises and the rule are combined to produce a new label for the conclusion. If the same conclusion is derived in multiple ways, their labels are aggregated.

Let L_{LP} be a logic-programming language (e.g., Horn clauses with negation-as-failure, or an extended rule language). A Labeled Logic Program is defined as:

$P=\langle \Sigma, G, M, R \rangle$ where:

- $\boldsymbol{\Sigma}$ is a set of *labeled facts* and *labeled rules*, including fact: $t{:}p(\bar{a})$ and rule: $t{:}(h \leftarrow b_1,\ldots,b_n, not\ c_1,\ldots,not\ c_m)$;
- G is a label algebra, providing a partial order or priority relation $<$, a compatibility predicate $F(t_1,\ldots, t_n)$, a propagation operator f, and an aggregation operator $\oplus$.
- M is a labeling discipline specifying how labels are introduced, transformed, and combined during inference.
- $\boldsymbol{\mathcal{R}}$ is the inference regime (e.g., SLD-resolution, stable-model semantics, or argument construction).

An inference rule in looks like the following:

- $t_r{:}(h \leftarrow b_1,\ldots,b_n)$
- $t_1{:}b_1,\ldots,t_n{:}b_n$
- $F(t_r, t_1,\ldots,t_n)$ holds

$$\overline{\mathrm{f}(t_r, t_1, \ldots, t_n){:}\, h}$$

 doi:10.20944/preprints202601.1549.v1

If multiple derivations of h exist, their labels are aggregated: $(t \oplus t'):h$. Notice that *the object-level rules remain unchanged*, while different logics emerge from different choices of G and M. To support multiple logical formalisms simultaneously, we use structured labels: $t=\langle t^{time}, t^{space}, t^{modal}, t^{provenance}, t^{argum}, t^{moda}, t^{prob}, t^{rel}\rangle$. Compatibility F and propagation f can be defined component-wise or with cross-constraints. Aggregation $\oplus$ may use:

- Max/min for reliability,
- Union for argument supports,
- Dempster–Shafer or interval intersection for probabilities,
- Set union for temporal supports.

This yields a *single* LLP that can be "projected" into different logics by selecting which components and constraints are active.

We define a *logic profile* π as a configuration that specifies:

- Which label components are active,
- Which constraints define compatibility F,
- Which propagation function f applies,
- Which aggregation $\oplus$ is used,
- Whether defeat/flattening is applied.

We now explain how to verify an LLM outputs with LLPs. Step 1 is claim extraction where LLM outputs are parsed into object-level literals such as $infection(p)$

Step 2 is label assignment. Each extracted claim is assigned a structured label:

$$t=\langle [t_1,t_2],\emptyset,w_0, \text{LLM}, support,[0.6,0.8],\emptyset\rangle$$

capturing time, source, argumentative role, and confidence.

Step 3 involves domain rules. A neutral object-level rule:

$$r:\ complication(x) \leftarrow infection(x), immunosuppressed(x)$$

The rule itself does not encode whether the reasoning is temporal, modal, or defeasible.

Step 4 performs multi-profile verification is based on:

- Temporal profile πtemp checks whether the infection occurred before the complication.
- Argumentation profile πarg considers exceptions (e.g., prophylactic treatment) that may defeat the rule.
- Probabilistic profile πprob propagates uncertainty and checks if confidence falls below a threshold.

Each profile yields a labeled verdict:

$$t1:supported(claim), t2:defeated(claim), t3:uncertain(claim)$$

Step 5 is meta-aggregation. These verdicts can themselves be aggregated into a final verification status: $t^*:unreliable(claim)$ where t^* explains *why* (temporal inconsistency, argumentative defeat, low probability).

By grounding verification in LLP inspired by Gabbay's LDS, we obtain a universal reasoning substrate for LLM outputs. The object-level logic program captures domain knowledge, while labels encode the meta-logical dimensions required for verification. Different logical formalisms emerge dynamically through label disciplines and profiles, enabling robust, explainable, and adaptive verification of LLM reasoning across heterogeneous domains.

Discourse and Claim Defeasibility

Hallucinations frequently arise when expliict rule (such as a typical diagnosis) does not hold. Labels in LLP bridge *rhetorical structure theory (RST)* and *symbolic reasoning*, enabling logical programs to dynamically adjust the *strength* of their rules based on the discourse hierarchy between *nucleus* and *satellite* components of a decision text. In medical, legal, or diagnostic narratives, these rhetorical

 doi:10.20944/preprints202601.1549.v1

relations capture how strongly a given statement supports the main conclusion, which can be encoded into logical inference or probabilistic weights.

Each rhetorical relation has a *nucleus* (the "main" proposition) and a *satellite* (supporting or contextual material). The satellite always carries less essential information than the nucleus (Table 1). One can see that nucleus contains main diagnostic/treatment fact (higher base probability) and satellite caries contextual/supporting info with lower significance. These values are obtained in the course of improvement of validation performance, described in Evaluation section.

Table 1. Relative weights of nucleus and satellite for different rhetorical relations.

Rhetorical Relation	Relative Argument Strength (Nucleus : Satellite)
Cause	0.8 : 0.2
Effect / Result	0.7 : 0.3
Condition	0.6 : 0.4
Contrast	0.55 : 0.45
Elaboration	0.65 : 0.35
Concession	0.75 : 0.25
Background	0.85 : 0.15
Enablement / Purpose	0.7 : 0.3
Evidence / Justification	0.6 : 0.4
Evaluation	0.65 : 0.35

The attenuation mechanism introduces *graded support* into inference by weighting premises according to their rhetorical role (Galitsky and Rybalov 2026)**.** For instance, in the "Cause" example:'

fever(patient) :- malaria_exposure(patient) [0.2].

fever(patient) :- high_fever(patient) [0.8].

the nucleus clause ("The patient developed a high fever") dominates inference, while the satellite clause ("Because the patient had recently returned from a malaria-endemic area") provides weaker contextual justification. During reasoning, if nucleus evidence is missing, the satellite's low weight prevents the rule from firing confidently. Conversely, if both are true, the conclusion is strengthened, but not absolutely certain.

In this way, attenuation acts as a defeasible weighting scheme inside the rule base: satellite conditions can be overridden when contradicted by stronger nucleus evidence. This mirrors defeasible reasoning (Antoniou & Billington, 2000) where less essential premises m ay fail without invalidating the entire argument.

In probabilistic logic programming, for example, ProbLog (De Raedt et al., 2007, Fierens et al. 2015) or LPADs (Riguzzi, 2018), rule attenuation becomes a numerical prior governing the probability of a rule firing. Each rhetorical relation translates into a weighted probabilistic clause, where the nucleus-to-satellite ratio (e.g., 0.8:0.2) determines the relative confidence of inference:

0.8::fever(patient) :- high_fever(patient).

0.2::fever(patient) :- malaria_exposure(patient).

Probabilistic inference then aggregates these weighted supports across multiple discourse relations to compute posterior probabilities for hypotheses (e.g., *pneumonia(patient), infection_cleared*). In this sense, rhetorical weighting becomes a proxy for epistemic strength: nucleus-driven rules act as high-confidence evidence, while satellite-driven rules introduce plausible but defeasible explanations

Hallucination in Health Dataset

We built upon the dataset for Autoimmune Disorders and Healthy Controls (Ragheb 2024) that serves as the foundation for generating a synthetic corpus of realistic clinical vignettes. It originates from structured clinical data representing 12,500 patients, covering a diverse spectrum of

 doi:10.20944/preprints202601.1549.v1

autoimmune disorders alongside healthy controls. The source data include detailed Complete Blood Count (CBC) parameters, key autoantibody markers, demographic attributes, and symptom profiles. Each autoimmune condition is characterized by disorder-specific autoantibody criteria aligned with established diagnostic standards, enabling reliable differentiation between disease states and normal baselines. Designed to support machine learning research in autoimmune diagnostics and prognostics, this structured dataset was alsoexpanded into narrative form to capture the variability and nuance of real-world clinical reasoning.

Our dataset contains 1,200 clinical vignettes designed to evaluate how large language models interpret and reason about nuanced patient narratives. We refer to it as *Autoimmune-narrate-halluc*. Each record includes a *health_complaint* field — a 2–5 sentence, fluent, natural, and grammatically correct first-person description of a patient's experience, written in authentic English with emotional realism and contextual detail (e.g., onset, duration, triggers, lifestyle impact). The accompanying *disease_description* field provides a concise 1–2 sentence hybrid explanation that combines layperson accessibility with clinical precision, summarizing typical presentation and diagnostic considerations. Together, these fields model the ambiguity, overlap, and conversational texture of real-world medical communication, offering a challenging yet controlled benchmark for hallucination detection and reasoning consistency in LLMs.

The snapshot of the dataset is available (https://anonymous.4open.science/r/halluc_in_health-733B/prolog/data/autoimmune_diseases_with_complaints.csv) and also the full dataset of 1200 complaints is available (https://anonymous.4open.science/r/halluc_in_health-733B/prolog/data/diseases_with_patient_complaints1000.xlsx).

Evaluation

We first evaluate on three claim-verification datasets that we derive from existing QA/NLI resources: TruthfulHalluc (from TruthfulQA; Lin et al., 2021), MedHalluc (from MedQA; Jin et al., 2020 and PubMedQA; Jin et al., 2019), and eSNLI_Halluc (from eSNLI; Camburu et al., 2018). For each source, we convert items into question–answer (QA) style pairs and then inject controlled inconsistencies by appending randomly sampled, semantically incompatible attributes (facts, circumstances, symptoms). These perturbations create positive "hallucination" cases; unmodified items serve as negatives. Our focus is hallucination detection for model answers using four logical assessment methods as validators. Each validator assesses whether an answer's central claim is defeated by the argument-validation system. We define a hallucination as a claim whose defeat probability exceeds 0.5. This cautious threshold is motivated by safety-critical domains (health, legal, finance), where we prefer to reject answers that are defeated with substantial probability.

Dataset size and prevalence are as follows. Each used hallucination dataset contains 1,000 QA pairs with a 2% hallucination rate. In the original source datasets the natural hallucination rate is <0.5%; our perturbation procedure raises prevalence to enable meaningful detection metrics and comparability with prior LLM-argumentation studies.

We then evaluate on our own dataset *Autoimmune-narrate-halluc* which is designed to cause hallucinations and make their detection as hard as possible. Hallucination rate exceeds 4%.

To aggregate evidence from multiple reasoning paradigms, we design a combination algorithm that integrates the outputs of four logical validators—logic programming (LP), probabilistic logic programming (PLP), argumentation, and abductive explanation—into a unified hallucination detection score. Each component independently assesses whether the claim inferred from the LLM's answer is defeated given the ontology and discourse context. The LP validator checks for explicit rule violations or missing entailments; the PLP validator estimates the posterior probability of the claim being supported given uncertain premises; the argumentation module computes whether the claim remains justified under admissible semantics; and the abductive module measures the minimal explanatory distance between the LLM's claim and the logically derivable one. Each produces a normalized score in [0,1] representing the probability of defeat (1 = fully defeated).

 doi:10.20944/preprints202601.1549.v1

The combination stage employs a weighted ensemble where the weight of each logic component depends on its historical reliability and discourse alignment. Specifically, weights are dynamically adjusted according to (a) the discourse role of the claim's nucleus and satellite segments, and (b) the inter-component agreement. When the nucleus of a discourse relation dominates, LP and argumentation receive higher weights (reflecting strict reasoning); when uncertainty or evidential justification prevails, PLP and abduction gain influence. LLP computes a defeat probability as a weighted mean of component outputs, applying rule attenuation from discourse relations (e.g., Cause 0.8:0.2) as priors.

Table 2 reports F1 scores for hallucination detection across four datasets using standard Logic Programming (LP), argumentation-based validation, and the proposed Labeled Logic Program (LLP) framework, with and without discourse-aware weighting (+d). Overall, LLP consistently outperforms traditional LP and argumentation on most benchmark datasets, demonstrating the benefit of labeling symbolic rules with semantic, epistemic, and contextual information.

Table 2. Hallucination prediction accuracy.

Dataset/ method	LP		arguments		LLP
discourse		+d		+d	
Truthful-Halluc	0.62	0.67	0.72	0.78	0.85
Med-Halluc	0.57	0.60	0.77	0.72	0.81
eSLNI-Halluc	0.58	0.62	0.68	0.67	0.69
Our dataset Autoimmune-narrate-halluc	0.39	0.37	0.32	0.35	0.33

On Truthful-Halluc, LLP achieves 0.72 without discourse modeling and improves to 0.78 with discourse cues, while the fully discourse-enhanced configuration reaches 0.85. Similarly, on Med-Halluc, LLP substantially improves over LP and argumentation, achieving 0.77 and further rising to 0.81 in the discourse-aware setting. These gains indicate that labeled rules enable more flexible and accurate validation of LLM-generated claims than unlabeled symbolic rules alone.

On eSNLI-Halluc, LLP provides moderate improvements over LP (0.68 vs. 0.58), reflecting the relatively simpler factual structure of the dataset, where fewer contextual and explanatory inferences are required. In contrast, the Autoimmune-narrate-halluc dataset remains challenging for all methods due to implicit symptoms, vague patient language, and contextual ambiguity. Here, LLP does not outperform simpler approaches, suggesting that highly narrative and underspecified medical text still exceeds the expressive capacity of current symbolic labeling schemes.

Overall, the results confirm that combining labeled symbolic reasoning with discourse-aware weighting yields the most robust hallucination detection performance on structured benchmarks, while highlighting open challenges for clinical narrative data. Overall, the results demonstrate that LLP offers a stronger and more general verification mechanism than classical LP or argumentation alone. Discourse-aware weighting (+d) further improves performance by prioritizing nucleus claims and attenuating weaker contextual statements, confirming the importance of rhetorical structure in hallucination detection. The highest accuracies on Truthful-Halluc (0.85) and Med-Halluc (0.81) indicate that combining labeled symbolic reasoning with discourse analysis provides a robust, interpretable, and scalable approach to LLM verification in safety-critical domains. The *Autoimmune-Narrate-Halluc* dataset formed in this study remains difficult: baselines are low (~0.4) and the combined systems modestly improve to 0.52. This reflects the challenge of fuzzy patient language, implicit symptoms, and contextual ambiguity not easily captured by formal rules.

Our MedHalluc results for argumentation are broadly comparable to prior work: ArgMed-Agents with GPT-4 reports 0.91 predictive accuracy (Hong et al., 2024); ArgLLM with GPT-4o reports 0.80 (Friedman et al., 2015); and an ensemble of ArgLLMs achieves 0.73 (Ng et al., 2025). That said, these systems estimate claim truthfulness, whereas our study predicts hallucination via whether a

 doi:10.20944/preprints202601.1549.v1

claim is defeated by the argument-validation module, so the targets differ and the numbers are not strictly comparable.

Human Evaluation Setting: Logic-Supported Trust Calibration

To complement the automatic metrics reported in Table 2, we conducted a controlled human evaluation to assess how logical verification tools can enhance the trustworthiness and interpretability of LLM outputs. The goal was to examine whether human evaluators, when assisted by formal reasoning modules, demonstrate improved accuracy and confidence in hallucination detection across four domains.

Twelve evaluators participated in the study, grouped according to domain expertise: (i) four biomedical professionals for MedHalluc, (ii) four computational linguists for eSNLI_Halluc, and (iii) four fact-checking specialists for TruthfulHalluc. Each participant evaluated 150 question–answer (QA) pairs sampled evenly from the three datasets, including both perturbed (hallucinated) and unmodified (factual) items.

The evaluation proceeded in three stages per item:

1. Baseline review — the evaluator inspected only the LLM answer and rated its factual soundness and confidence on a 0–5 Likert scale.
2. Logic-assisted review — the evaluator was shown the logical verification report generated by the four reasoning modules (LP, PLP, Argumentation, Abduction) and the combined discourse-aware ensemble.
3. Confidence re-rating — the evaluator revised the initial confidence score and provided short written feedback on interpretability and explanatory clarity.

The logic-support interface presented the following information for each answer:

- Defeat probabilities for LP, PLP, Argumentation, Abduction, and their ensemble.
- Textual explanations derived from symbolic traces, e.g., "Claim 'Fever and ankle pain indicate gout' is defeated (0.72): rule [Gout → joint pain, swelling] not satisfied; missing causal link fever → gout."
- Discourse weighting view, where nucleus–satellite relations were visualized as strength attenuations (e.g., Cause 0.8 : 0.2). This format enabled evaluators to see why a statement was marked as inconsistent and which rule or discourse segment contributed most to defeat.

We measure both quantitative and qualitative outcomes in Table 3.

Table 3. Metrics of human evaluation.

Metric	Definition
Accuracy	Percentage of correct hallucination judgments by humans.
ΔConfidence	Mean change in confidence after seeing logical explanations.
Human–logic agreement (κ)	Cohen's κ between final human decision and ensemble output.
Calibration error	Difference between human confidence and true correctness.
Interpretability rating	Self-reported clarity of the logical explanation (1–5 scale).

The aggregate results (Table 4) show consistent gains across all logic-assisted conditions. The results indicate that logical verifiers significantly improve both accuracy and subjective trust. Participants reported that reasoning traces helped them *understand* system behavior rather than merely accept or reject outputs. Agreement with the ensemble's defeat probabilities correlated with higher interpretability ratings ($r = 0.72$). Notably, discourse-aware weighting yielded the largest trust gain, suggesting that humans find explanations framed in rhetorical terms (nucleus vs. satellite) especially intuitive.

 doi:10.20944/preprints202601.1549.v1

Overall, this human-in-the-loop experiment demonstrates that logic-based validation not only detects hallucinations but also *humanizes* verification: it enables users to perceive LLM reasoning as accountable and auditable, bridging statistical generation with symbolic justification.

Table 4. Evaluation of accuracy, confidence and interpretability.

Condition	Human accuracy	ΔConfidence	Agreement (κ)	Interpretability (1–5)
LLM only	0.67	—	—	2.3
+ LP	0.73	+0.12	0.54	3.1
+ PLP	0.74	+0.14	0.56	3.4
+ Argumentation	0.75	+0.16	0.59	3.7
+ Abduction	0.71	+0.11	0.51	3.3
C (d-aware)	0.83	+0.22	0.68	4.2

Implementation

We describe a hybrid neuro-symbolic diagnostic reasoning pipeline where an LLM and a reasoning engine work together to verify or refute an LLM-generated decision (Figure 2). The goal of the architecture is to ensure that an LLM's diagnostic answer (for example, "The patient has gout") is not only linguistically plausible but also logically justified and consistent with available structured ontology constructed by the LLM on demand. If the logical reasoning pipeline cannot defeat the LLM diagnosis claim, it is confirmed; otherwise, the LLM answer is marked unconfirmed.

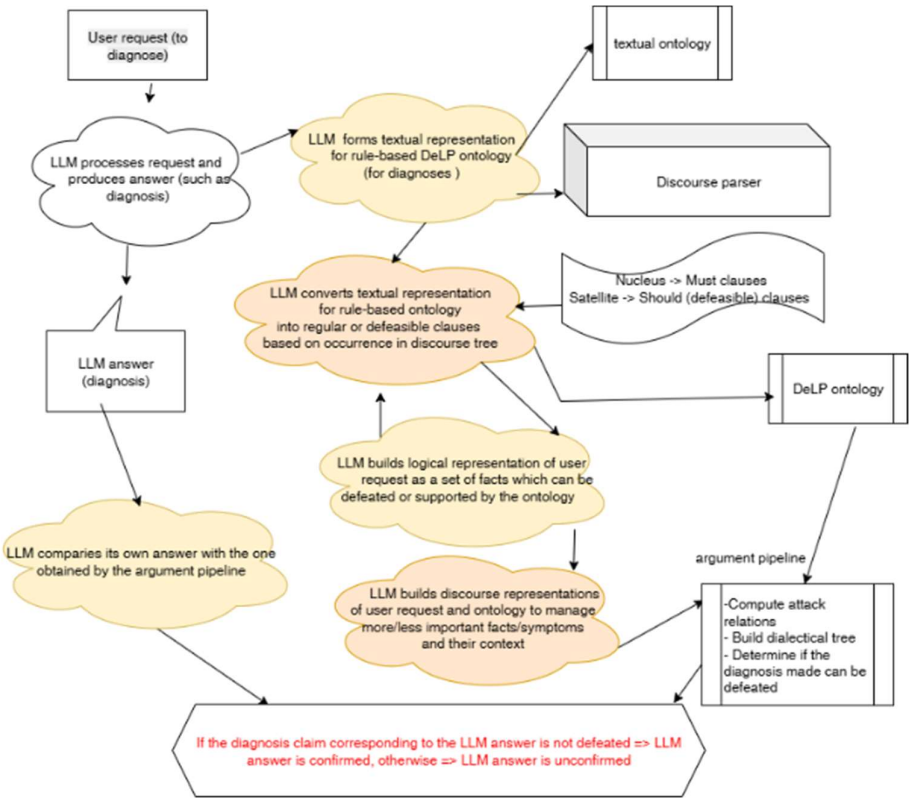

Figure 2. System architecture.

The ValidLog4LLM's workflow is as follows:

1. User input. The process begins with a user request, such as asking ValidLog4LLM to provide a diagnosis.

 doi:10.20944/preprints202601.1549.v1

2. LLM generates initial answer. The LLM processes the request and outputs an initial diagnosis or conclusion (e.g., "The disease is gout").
3. Ontology and discourse setup. A textual ontology of medical knowledge (rules, relationships, symptoms, conditions) and a discourse parser are available. The discourse parser identifies rhetorical relations in the text — for instance, nucleus (main facts) and satellite (contextual or defeasible facts). Nucleus → "Must" clauses (non-defeasible rules) and Satellite → "Should" clauses (defeasible rules)
4. LLM forms ontology representation, transforming textual information into a rule-based ontology in the DeLP format — essentially translating natural-language reasoning into structured logical rules.
5. Conversion to logic program. The LLM converts these rules into regular (strict) or defeasible (soft) clauses depending on their role in the discourse (main vs. secondary information).
6. Building logical representation of the user request. The system formalizes the user's question and the LLM's proposed answer as a set of logical facts that can either be defeated or supported by the ontology.
7. Discourse representation integration. The LLM builds discourse representations of both the user request and the ontology, capturing which arguments are more or less important (nucleus/satellite weighting) and how they relate contextually.
8. Argumentation pipeline. The argumentation module computes attack relations among rules (contradictions or counter-arguments), dialectical trees, representing possible argumentative dialogues between supporting and opposing claims, and defeasibility outcomes, determining whether a claim survives all counter-arguments
9. Comparison and validation. The LLM compares its original diagnosis with the verified diagnosis obtained through the logical inference process.
10. Decision. If the logical reasoning shows that the diagnosis claim is not defeated, it is confirmed as valid. If the claim is defeated by stronger counter-arguments from the ontology, it is marked unconfirmed.

Conclusions

We presented a hallucination detection framework that integrates multiple reasoning paradigms within a Labeled Logic Program (LLP) architecture. Experiments on three benchmark datasets show that LLP consistently outperforms classical logic programming and argumentation-based validation, especially when combined with discourse-aware weighting that prioritizes nucleus claims over peripheral statements. The results highlight the importance of labeling rules with semantic and contextual information for robust verification of LLM outputs. While performance remains limited on highly narrative medical data, the framework provides interpretable explanations of logical defeat. A human evaluation further confirms that logic-supported verification improves both accuracy and user trust, with discourse-aware explanations perceived as especially intuitive. Overall, LLP offers a scalable and interpretable approach to LLM verification in safety-critical domains.

Code: https://github.com/bgalitsky/halluc_in_health

Datasets: https://github.com/bgalitsky/halluc_in_health/tree/master/data

References

Gabbay DM and Woods J. Labelled Deductive Systems, Editor(s): Dov M. Gabbay, John Woods. A Practical Logic of Cognitive Systems, Elsevier, V1, 2003, pages 369-394,

Galitsky, B.; Rybalov, A. Neuro-Symbolic Verification for Preventing LLM Hallucinations in Process Control. Processes 2026, 14, 322. https://doi.org/10.3390/pr14020322

Kalai AT Why Language Models Hallucinate. 2025 arXiv:2509.04664

 doi:10.20944/preprints202601.1549.v1

Ragheb A. Comprehensive Autoimmune Disorder Dataset. https://www.kaggle.com/datasets/abdullahragheb/all-autoimmune-disorder-10k. 2024.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence, 77(2), 321–357.

Fierens, D., Van den Broeck, G., Renkens, J., Shterionov, D., Gutmann, B., Thon, I., Janssens, G., & De Raedt, L. (201). Inference and learning in ProbLog. Theory and Practice of Logic Programming, 15(3), 358–401.

Lin S, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. CoRR, abs/2109.07958, 2021.

Jin, B. Dhingra, Z. Liu, W. W. Cohen, and X. Lu, PubmedQA: A dataset for biomedical research question answering," 2019.

Jin D, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. CoRR, abs/2009.13081, 2020.

Hong S, Liang Xiao, Xin Zhang, Jianxia Chen (2024) ArgMed-Agents: Explainable Clinical Decision Reasoning with LLM Disscusion via Argumentation Schemes

Camburu O-M, Tim Rocktäschel, Thomas Lukasiewicz, Phil Blunsom (2018) e-SNLI: Natural Language Inference with Natural Language Explanations. Advances in Neural Information Processing Systems 31 (NeurIPS 2018)

De Raedt, L., Kimmig, A., & Toivonen, H. (2007). ProbLog: A probabilistic Prolog and its application in link discovery. IJCAI, p 2468.

Riguzzi, F. Foundations of probabilistic logic programming. River Publishers. New York. 2022.

Freedman, Gabriel & Dejl, Adam & Gorur, Deniz & Yin, Xiang & Rago, Antonio & Toni, Francesca. Argumentative Large Language Models for Explainable and Contestable Claim Verification. Proceedings of the AAAI Conference on Artificial Intelligence. 2025. 39. 14930-14939

Ng, Ming & Jiang, Junqi & Freedman, Gabriel & Rago, Antonio & Toni, Francesca. MArgE: Meshing Argumentative Evidence from Multiple Large Language Models for Justifiable Claim Verification. 2025. 10.48550/arXiv.2508.02584.