Integrating Language Models into Inductive Logic Programming: Enhancing Knowledge Integration and Human-Centric Explainability

Sheila Favaedi[1,2*,†], Shiva Favaedi[1,2*,†], Aqib Hafiz[1,2*,†], Harsh Marthak[1,2*,†], Ali Shahebrahimi[1,2*,†], Kishore Krishna Srinivasan[1,2*,†], Vedat Yasar[1,2*,†], Graeme Gourlay[1], and Alireza Tamaddoni-Nezhad[2]

[1]Azaries Ltd., The Technology Centre, Surrey Research Park, Guildford, GU2 7YG, UK
[2]University of Surrey, UK

{vedat.yasar, kishore.srinivasan, harsh.marthak, sheila.favaedi, shiva.favaedi, aqib.hafiz, ali.shahebrahimi, graeme.gourlay}@azaries.com, a.tamaddoni-nezhad@surrey.ac.uk

Abstract. Inductive Logic Programming (ILP) enables knowledge-driven learning and interpretability by generating symbolic rules that make machine learning decisions transparent. However, ILP relies on expert-generated or pre-defined encoding of background knowledge. Moreover, its output, while formally explainable, can be difficult for humans to read and interpret, limiting their practical utility. We propose a hybrid framework in which LLMs are integrated with ILP both to generate logic-compatible knowledge representations from raw data and to verbalise symbolic rules into natural language. These enhancements are treated independently, allowing us to evaluate their distinct contributions to explainability and learning performance. Through empirical studies grounded in real-world datasets, we explore the potential of language models not only as surface-level communicators, but as active participants in symbolic reasoning workflows. The rule translation task focuses on whether language models can express logic clauses in ways that are faithful to their original structure but significantly easier for humans to interpret. In parallel, we compare ILP performance using LLM-generated background knowledge against pre-defined baselines. Our findings suggest promising directions for combining the strengths of symbolic logic and neural generation to build AI systems that are both formally grounded and accessible to a broader range of users.

Keywords: Inductive Logic Programming (ILP) · Large Language Models (LLMs) · Explainable AI · Neural-Symbolic Methods · Rule-Based Learning · Background Knowledge Induction · Natural Language Explanation · Human-Centered AI

1 Introduction

Artificial Intelligence (AI) continues to make remarkable progress in various domains, particularly through large-scale neural architectures such as Large Language Models (LLMs). These models demonstrate fluency in text generation, summarization, translation, and increasingly, reasoning tasks. However, as their influence spreads into safety-critical areas such as healthcare, finance, autonomous systems, and law, the demand for transparent and verifiable AI systems is intensifying. While LLMs are powerful at producing plausible text, they remain limited in their ability to provide verifiable reasoning paths. This has led to a resurgence of interest in symbolic approaches that offer formal rigor and traceability, especially **Inductive Logic Programming (ILP)**.

Inductive Logic Programming (ILP) [22,23] stands out as one of the few machine learning approaches that inherently delivers knowledge-driven and human-interpretable outputs. Unlike neural networks, which often operate as opaque black boxes, ILP learns logic-based rules that explicitly model reasoning over structured data. These symbolic rules offer transparency, traceability, and the ability to verify decisions—a powerful advantage in high-stakes domains such as healthcare, finance, and legal systems. However, despite these strengths, ILP remains underutilised in applied settings. The reason lies not in its capability, but in its reliance on expert-encoded background knowledge and the communicability of its output. ILP rules are typically expressed in formal logical syntax, which can appear arcane, inaccessible, or even misleading without

† These authors contributed equally to this work. Author names are listed in alphabetical order.

proper contextualisation, even to technically literate stakeholders. As a result, many domains that would benefit from logic-based, interpretable AI often bypass ILP in favor of more familiar, albeit opaque, statistical models.

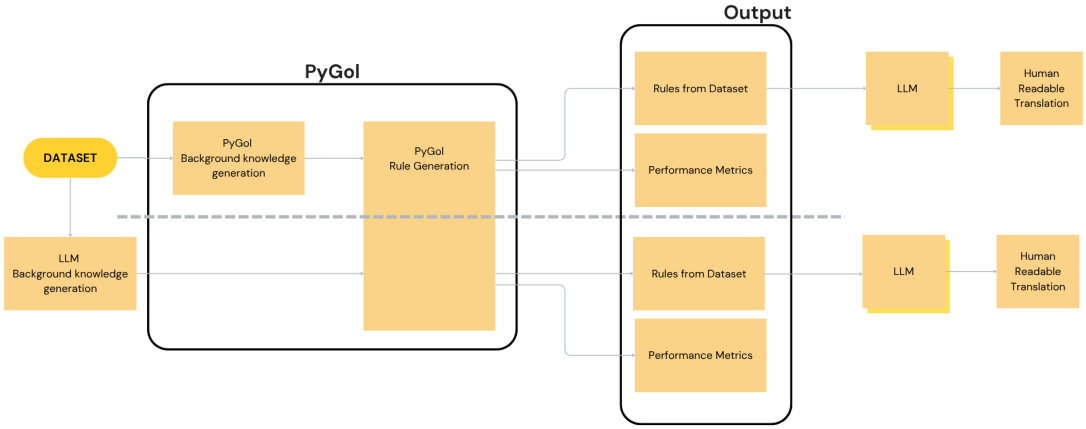


Fig. 1: System architecture of the proposed hybrid ILP+LLM framework. The dataset serves as input to both PyGol and a large language model (LLM) for background knowledge generation. Two parallel pipelines are evaluated: one using pre-defined PyGol-generated background knowledge, and the other using LLM-generated knowledge. In both cases, the resulting rules and performance metrics are processed by an LLM to generate human-readable explanations, enabling comparative evaluation of interpretability and performance.

These barriers represent a significant bottleneck to the adoption of ILP. In modern AI systems, where explainability and trustworthiness are increasingly essential—both from regulatory and ethical perspectives—the inability to translate symbolic logic into human-understandable explanations limits the adoption of ILP solutions. Decision-makers, domain experts, and interdisciplinary researchers require clear, natural language justifications to interpret, verify, and trust automated outputs. Without such accessibility, the value of ILP—despite its strong theoretical underpinnings—remains confined to a small subset of specialists. As Cropper et al. [10] bluntly observe, "ILP systems are still notoriously difficult to use and you often need a PhD in ILP to use any of the tools." When even ILP researchers face usability challenges, the barrier for wider academic and industrial uptake becomes nearly insurmountable. To address this, the community must prioritize the development of standardized, user-friendly, and robustly engineered tools.

At the same time, LLMs have demonstrated emergent capabilities at scale [5,32]. Techniques such as chain-of-thought prompting and instruction tuning have enabled LLMs to break complex tasks into step-by-step reasoning paths, mimicking structured logic. However, these capabilities remain largely "soft"—they lack the formalism, consistency, and provability that define symbolic systems. Furthermore, their internal decision-making processes are often inscrutable. This has prompted several studies to explore the potential of **neural-symbolic integration**, where symbolic systems like ILP are paired with neural models like LLMs to combine the best of both worlds: rigor and fluency, logic and language [6,21].

Recent efforts to align language models with symbolic reasoning include transformer models trained on rule-based corpora [39], hybrid architectures like Knowledge Graph of Thoughts [2], and logic-augmented generation frameworks such as RuAG [41]. While these approaches reveal the promise of logic-infused LLMs,

they often treat symbolic systems as downstream or auxiliary modules. ILP, in particular, remains underrepresented in these efforts, despite its demonstrated strengths in generalisation from small data, interpretability, and support for abductive and inductive reasoning.

The significance of ILP in AI safety and scientific discovery has already been established through applications in drug design, ecological modeling, and explainable robotics [29,9,10]. However, its lack of integration with current LLM pipelines has prevented wider adoption. There is a critical need for systems that do not merely use ILP for rule learning or explanation post-processing but treat ILP and LLMs as collaborative agents in both the reasoning and communication process.

While LLMs show strong linguistic fluency, their reasoning capabilities remain limited. Betz et al. [3] used synthetic arguments to train GPT-2 in deductive reasoning, but the models struggled with inductive inference. Clark et al. [7] framed transformers as soft reasoners, lacking structured logic, while the HuggingFace library [34] facilitated widespread LLM experimentation. In theorem proving, Polu and Sutskever [26] fine-tuned GPT-f on Metamath, achieving state-of-the-art results, and Wang [31] introduced MetaGen to generate synthetic theorems, improving prover generalisation. Evaluation studies further question LLM reasoning: Xu et al. [36] found unstable performance in deductive and abductive tasks; Yeo et al. [38] revealed the limits of chain-of-thought prompting; and Wu et al. [35] introduced a counterfactual framework exposing overfitting to training heuristics. To support more rigorous reasoning, Lu et al. [18] proposed MathGenie for generating verifiable maths problems, while Morishita et al. [20] introduced FLD, a formal logic corpus for deductive training. Together, these works highlight the promise of domain-specific data and symbolic integration for advancing LLM reasoning.

We present a hybrid framework in which LLMs serve dual roles: first, as generators of domain-specific background knowledge (BK) to support ILP's learning process; and second, as translators that convert ILP-generated rules into clear, accurate, natural language explanations. Through this approach, we aim to close the interpretability gap that has long hindered the practical adoption of symbolic AI.

In doing so, we contribute to a growing line of work that treats interpretability not as a final-stage add-on, but as an integral part of the reasoning architecture. Our results in theory evaluation that combining ILP and LLMs can produce systems that are not only accurate but also intelligible to non-specialists. This alignment of **symbolic rigor with linguistic clarity** could be essential for the next generation of AI systems: ones that reason reliably, learn adaptively, and communicate understandably. This alignment is further clarified through a comparative summary of reasoning approaches (see Table 1).

Table 1: Comparison of Reasoning Approaches

Feature	**Symbolic (ILP)**	**Neural (LLMs)**	**Hybrid (ILP + LLM)**
Interpretability	High	Low	High
Generalisation	Strong	Weak	Preserved
Reasoning	Logic-based	Limited	Logic-based
Background Knowledge	Pre-defined	Not structured	LLM-generated
Adaptability	Low	High	Moderate–High
Explainability	Formal	Limited	Human-centered

2 Related Work

2.1 Symbolic Reasoning and ILP's Position

ILP has long been valued for its ability to learn logic-based rules from small amounts of data, offering natural support for deduction, induction, and abduction. Its interpretability and formal structure make it ideal for applications in scientific discovery, bioinformatics, and explainable AI [22,10]. Notably, Chaghazardi et al. [6] applied ILP in traffic sign detection, demonstrating that systems like Aleph and Metagol could achieve 100% adversarial robustness using few-shot logic rule induction. This underscores ILP's strength in safety-critical domains where transparency is essential.

However, symbolic logic alone is insufficient for complex, noisy, real-world data. ILP systems require structured background knowledge that cannot always be generated from raw inputs. Hence, ILP usability remains limited; as noted by Cropper et al. [10], most ILP tools require expert-level understanding just to operate. Despite their power, these systems remain confined to academic niches unless integrated into more accessible learning architectures.

2.2 Language Models and the Limits of Neural Reasoning

In contrast, recent LLMs such as GPT-3, GPT-4, and PaLM exhibit remarkable linguistic abilities and can perform reasoning tasks when guided with techniques like chain-of-thought prompting [32]. Yet their reasoning remains stochastic and opaque. While these models can generate convincing text that resembles logical reasoning, they lack the guarantees of soundness and completeness inherent in formal logical approaches such as ILP.

This limitation is evident in benchmarks like ART, where even state-of-the-art models like BERT and GPT-2 scored well below human performance in abductive inference tasks [4]. Similarly, Liu et al. [17] introduced LogiQA 2.0 to benchmark logical reasoning in reading comprehension and found that models such as GPT-3 reached only 68.65% accuracy—far below the human baseline of 89.36%. FOLIO [15] extended this challenge to first-order logic, revealing that current LLMs struggle with semantic generalisation despite achieving syntactic validity.

Studies such as Gontier et al. [14] and Tafjord et al. [28] confirm that LLMs can generate structured proofs or explanations in natural language, but these outputs often fail under length-based generalisation, recursion, or noisy data. Thus, while LLMs can mimic reasoning, they often rely on surface-level patterns rather than logic-based inference paths.

2.3 Toward Neural-Symbolic Integration

Bridging this gap has become a major focus of hybrid AI research. Frameworks such as CoLM [37], Logic-LM, and LogicAsker [40] propose chaining LLMs together to simulate modular reasoning steps. The RuAG framework combines Monte Carlo Tree Search (MCTS) with LLM outputs to induce logic rules for tasks like relation extraction and decision-making [41]. Meanwhile, the Knowledge Graph of Thoughts (KGoT) architecture builds task-specific symbolic graphs from LLM outputs, enabling multi-step procedural reasoning [2]. These efforts mark substantial progress, but they generally stop short of explicitly integrating formal systems like ILP into the reasoning process.

Only a few works have explicitly used ILP as a logic engine in neural-symbolic pipelines. Varghese et al. [30] proposed Meta Inverse Entailment (MIE), which leverages the ILP system PyGol to generate grammar rules that improve LLM performance in formal reasoning. Gandarela et al. [13] use a more formalised approach with an expressivity-graded evaluation framework, combining LLMs and the ILP system Popper in an iterative loop to induce and validate logical theories across synthetic datasets. Their results show that LLMs can match ILP systems like Popper in accuracy under high-noise conditions, but still struggle with rule chaining and recursion—highlighting both promise and limitations.

Additionally, background knowledge (BK) acquisition remains a key bottleneck in ILP. Afroozi Milani et al. [1] explore commonsense knowledge injection from ConceptNet, while Tamaddoni-Nezhad et al. [29] and Cornelio & Thost [8] propose synthesising BK through ILP-based scientific modeling and synthetic data generation, respectively. These efforts indicate a trend toward **automated BK generation**, a direction our work builds upon.

2.4 Human-Centered Explainability and ILP Translation

Equally important is the downstream challenge of explanation. While LLMs can generate fluent justifications, they may fail to reflect the actual reasoning steps used by underlying symbolic models. Ignatiev et al. [16] introduced an abductive explanation framework using SMT and MILP solvers to provide subset-minimal,

formally valid justifications for model decisions. Such efforts reflect a growing demand for **verifiable explainability**, especially in regulated or high-stakes domains.

Despite this, very few systems have attempted to translate ILP-generated rules into human-readable natural language in a systematic, user-evaluated way. This translation gap impedes ILP's adoption by non-experts and limits its broader utility. Existing LLM-based explanation tools focus more on shallow summaries or linguistic paraphrasing, without preserving logical fidelity.

In summary, while ILP provides unmatched clarity and generalisation with formal soundness guarantees, it remains underleveraged in modern AI pipelines. Conversely, LLMs dominate reasoning benchmarks but struggle with logical grounding. The literature suggests a promising space for hybrid frameworks that integrate ILP's symbolic precision with LLMs' generative fluency—especially when paired with human-centered evaluation. Our work builds on these foundations to develop a system that positions LLMs as both **knowledge assistants** and **explanation translators** for ILP, advancing the interpretability and usability of logic-based AI systems.

3 Problem Definition

While symbolic AI approaches such as ILP and generative AI models such as Large Language Models (LLMs) have evolved in parallel, their integration remains under-explored—especially in the context of explainability and structured learning. This disconnect exposes a two-sided bottleneck: symbolic models struggle with accessibility and usability, while language models lack formal reasoning control and verifiability. As AI systems become more widely adopted in critical domains, addressing this divide becomes essential.

Despite ILP's proven strengths in rule generalisation, recursive logic handling, and abductive inference, its practical deployment is constrained by three primary challenges:

1. **Inaccessibility of ILP Representations**: The outputs of ILP systems—Prolog-style logic rules—are highly formal and require significant expertise to interpret. While ideal for developers and computational logic experts, they remain obscure to domain professionals, stakeholders, and interdisciplinary users who must rely on these outputs to make high-stakes decisions [10].
2. **Manual Bottlenecks in Background Knowledge (BK) Engineering**: ILP performance heavily depends on the relevance and structure of background knowledge. However, defining high-quality BK typically requires deep domain knowledge and logical formalism, creating a labor-intensive barrier to entry. As noted by Cornelio and Thost [8], even synthetic environments struggle to model noise and structure trade-offs realistically without extensive manual design.
3. **Lack of Natural Language Alignment**: While LLMs are capable of generating fluent, context-sensitive explanations, they do not natively operate over symbolic representations. This means that outputs from ILP cannot easily be fed into LLMs for end-user consumption without bespoke translation pipelines—something most ILP tools currently lack.

The literature offers several pieces of this puzzle. Some works explore automated BK generation[1,13], while others investigate LLM reasoning benchmarks [4,17] or formal explanation models using SMT or MILP solvers [16]. However, there is no unified framework that positions LLMs as **dual agents** in the ILP pipeline: first, to assist in generating BK; second, to render ILP outputs in readable, logically faithful natural language.

We define this research gap as the lack of a hybrid reasoning pipeline in which LLMs support ILP both upstream (BK generation) and downstream (rule translation), while preserving logical integrity and human interpretability. Our objective is to design and evaluate such a pipeline by asking two guiding research questions:

- **RQ1:** To what extent can Large Language Models (LLMs) generate structured and relevant background knowledge for ILP systems?
- **RQ2:** Can LLMs reliably translate ILP-generated rules into natural language explanations that are perceived as clearer and more insightful by non-expert users?

Addressing these questions has both theoretical and practical implications. Theoretically, it enables a better understanding of the limitations and complementarities between symbolic and neural representations. Practically, it reduces the manual burden of ILP system design and opens its interpretability benefits to a wider audience.

This problem space motivates our methodological approach: an iterative, graded evaluation of LLM contributions to both knowledge induction and human-centered explanation within ILP-driven AI.

4 Methodology

To address the practical and interpretability challenges of symbolic learning systems, we propose a dual-role integration of large language models (LLMs) within an ILP framework. Our methodology is designed around two complementary objectives: first, to translate ILP-generated logic rules into natural language without losing logical fidelity; and second, to explore the capacity of LLMs to generate background knowledge (BK) that meaningfully supports ILP rule induction. Both tasks are conducted using PyGol as the underlying ILP engine. This integration allows us to assess not only the internal performance of ILP in logic learning, but also the external usability of its outputs through language-based explanations. The overall framework is evaluated through rule-level comparisons across four instruction-tuned LLMs—Google T5 [27], DeepSeek R1 Distill Qwen 7B [12], Mixtral-8x7B-Instruct [19], and GPT-4o [25]—and through targeted case studies in biomedical domains including Breast Cancer diagnosis [33] and drug-related mutagenicity classification [11].

4.1 System Overview

This work introduces a modular framework that enhances Inductive Logic Programming (ILP) by embedding large language models (LLMs) at two strategically important stages: one to improve how symbolic outputs are communicated, and another to support the learning process itself. At its core, the pipeline relies on PyGol for logic rule induction, using structured inputs and labeled examples to generate Prolog-style hypotheses that satisfy standard ILP criteria for completeness and consistency. To improve interpretability, we integrate a human readable explanation component in which LLMs translate symbolic rules into accessible natural language. This step is critical for enabling broader usability of ILP systems, particularly in domains like medicine and biology, where rule comprehension by non-technical users is essential. Parallel to this, we investigate background knowledge generation using LLMs, asking whether models like Mixtral can extract structurally useful logic templates from raw tabular data, thereby alleviating one of ILP's most significant bottlenecks: the manual encoding of domain knowledge. Both enhancements are evaluated independently. Explanation quality is measured through human judgment and user preference studies, while the utility of generated background knowledge is quantified through PyGol's classification accuracy and F1 scores.

Although these components are architecturally compatible and could support a unified looped reasoning pipeline, this study treats them as isolated modules to allow focused investigation of each improvement. The next two sections present these components in detail, beginning with human readable explanation of symbolic rules, followed by background knowledge generation and its impact on ILP performance.

4.2 Translating ILP Rules into Natural Language

One of the primary obstacles to the widespread adoption of Inductive Logic Programming (ILP) systems is not the accuracy of the induced rules but the difficulty users face in understanding their symbolic representations. Although ILP engines such as `PyGol` produce logically valid and high-performing hypotheses, these are often expressed as Prolog-style clauses that are largely inaccessible to users without formal training in logic programming. This lack of interpretability is especially problematic in domains such as medical diagnostics and chemical informatics, where domain experts must evaluate and make decisions based on model outputs.

To address this challenge, we explore whether large language models (LLMs) can serve as effective semantic translators of ILP rules. Our goal is to convert symbolic clauses into fluent, natural language explanations that maintain the logical structure and domain semantics of the original rules, thus making them comprehensible to users without programming expertise.

In our framework, ILP rules are generated using `PyGol` without any simplification or preprocessing. To enable accurate and contextually grounded translations, each ILP rule is paired with a standardized prompt containing contextual information about the dataset and task. This context includes a general description of the dataset, detailed explanations of relevant features or attributes, and illustrative examples. While the overall prompt structure remains consistent, the specific contextual content varies depending on the dataset to reflect domain-specific characteristics.

We evaluate three state-of-the-art LLMs: GPT-4o, Mixtral-8x7B-Instruct-v0.1.Q4_K_S, and DeepSeek-R1-Distill-Qwen-7B-Q4_K_M. Each model receives the same prompt template that instructs it to translate the provided ILP rule into clear, human-readable language. The instructions emphasise generating as detailed and simplified an explanation as possible without losing any factual or logical information. Additionally, models are implicitly guided to preserve numerical values, units, and critical details to maintain logical fidelity.

Our experiment encompasses a total of fourteen ILP rules, twelve drawn from the breast cancer dataset and two from the Mutagenicity dataset. These rules were selected without any filtering or additional criteria, representing a straightforward sample of the ILP system's output.

To evaluate the quality of the translations, we conduct a user study involving both domain experts and students familiar with the relevant fields. Participants are presented with each symbolic ILP rule alongside natural language explanations generated by all three models. The outputs are displayed side-by-side, with model identities clearly labeled. For each set, participants select the best translation and provide ratings on three dimensions: clarity, logical fidelity, and domain relevance. The evaluation combines comparative ranking and Likert-scale ratings to capture nuanced preferences. We also collect timing data on how long participants take to read and evaluate each explanation, providing insights into the cognitive load and efficiency associated with different models' outputs.

Figures 2 and 3 showcase representative examples where the symbolic ILP rule is paired with natural language explanations produced by the LLMs. These examples illustrate the ability of the models to incorporate domain-specific terminology and logical nuance while remaining accessible.

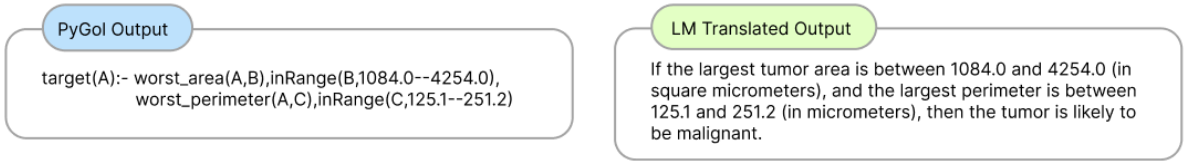


Fig. 2: Example of translating an ILP rule from `PyGol` into natural language. The left box shows the symbolic clause; the right shows the corresponding LLM-generated explanation.

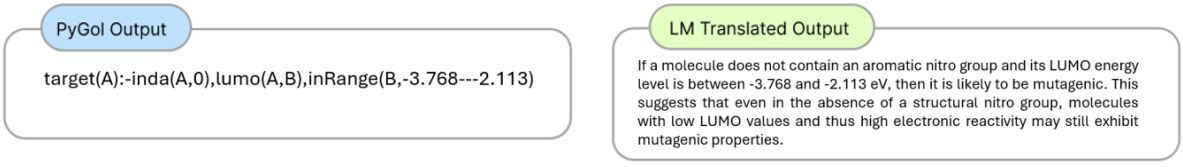


Fig. 3: Another example of ILP-to-natural language translation. The left box shows the symbolic clause; the right shows the corresponding LLM-generated explanation.

The Mutagenicity dataset, characterised by complex relational rules describing molecular substructures, provides a rigorous test of the models' ability to accurately convey graph-based reasoning in natural language. These rules involve specific chemical predicates such as the presence of nitro groups adjacent to aromatic

rings or the positioning of halogen atoms along molecular chains. The dataset's domain specificity demands precise preservation of chemical vocabulary and logical conditions, challenging the models to generate both semantically accurate and contextually meaningful translations.

Our results reveal that the models exhibit complementary strengths. Some are particularly effective at preserving exact numerical thresholds and logical operators, while others produce more fluent and accessible descriptions but occasionally omit or simplify critical logical conditions. Timing data suggests differences in reading and comprehension efficiency among the outputs, indicating varying cognitive loads depending on the model used. Rather than producing a single ranking of overall model performance, we analyse these differences to understand how model architecture, scale, pretraining corpus, and instruction tuning influence their ability to translate symbolic logic into natural language faithfully.

This work addresses a fundamental question in explainable artificial intelligence: whether large language models can generate natural language explanations of symbolic ILP rules that are both logically faithful and accessible to domain experts without programming backgrounds. Our findings suggest that with appropriate contextual prompting, neural-symbolic translation offers a promising pathway to making ILP systems interpretable and usable in real-world expert decision-making pipelines (see Figures 4 and 5).

4.3 LLM-Generated Background Knowledge (BK)

The second component of our methodology investigates whether large language models (LLMs) can generate background knowledge (BK) that is structurally sound and practically useful for rule induction in ILP. Unlike traditional ILP systems such as `Aleph` which requires pre-encoded background knowledge or `PyGol`, which has a simple pre-defined converter to derive relational BK directly from structured data, we explore a comparative setup where BK is instead generated by an LLM. This allows us to examine the potential of LLMs not only as explainers but also as upstream reasoning collaborators in the ILP pipeline.

We selected the *Breast Cancer* dataset as the experimental base. Initially in tabular format, each patient instance was converted into a logical form using feature-value predicates. To generate BK, we employed `Mixtral-7B-Instruct`, an instruction-tuned open-weight model known for its efficient domain adaptation and Prolog-compatible output. Mixtral was chosen for its interpretability and syntactic precision, critical for generating logic-ready BK compatible with ILP engines.

Our first attempt involved prompting Mixtral directly with raw CSV rows, asking it to group numerical values into statistical quartiles (e.g., g0 to g3). However, this revealed a key limitation: LLMs lack the ability to observe entire column distributions due to context window constraints. This led to inconsistent labeling of identical values across prompts and occasionally conflicting facts for single features.

To address this, we adopted a hybrid approach. All discretisation of numerical attributes was performed externally using Python's `pandas.qcut()`, which guarantees stable and globally consistent binning. The resulting discrete feature-value pairs were then formatted into Prolog syntax by Mixtral. This division of labor ensured statistical consistency while utilising the LLM's strength in structured output generation. Each data chunk was processed independently, and all prompt-response interactions were logged for traceability. The final LLM-produced file, `BK_LLM.pl`, mirrored the structure of `PyGol`'s native `BK.pl`.

We then constructed labeled positive and negative examples for ILP in standard format, ensuring compatibility with `PyGol`'s completeness (coverage of all positives) and consistency (exclusion of negatives) checks during hypothesis induction.

5 Results and Evaluation

Our approach is motivated by the need to make ILP outputs more accessible to non-experts, building on prior work such as Muggleton et al. [24], who defined and empirically tested the comprehensibility of ILP-learned programs through participant-based studies. Their experiments demonstrated that even when humans cannot independently induce symbolic rules, they can apply ILP-generated definitions effectively once provided. This aligns with our goal of enhancing ILP explainability through natural language translation and human interpretability.

Fig. 4: ILP rule breakdown and human-readable explanation

ILP Rule

```
target(A) :-
worst_area(A, B),
inRange(B, 1084.0-4254.0),
worst_perimeter(A, C),
inRange(C, 125.1-251.2).
```

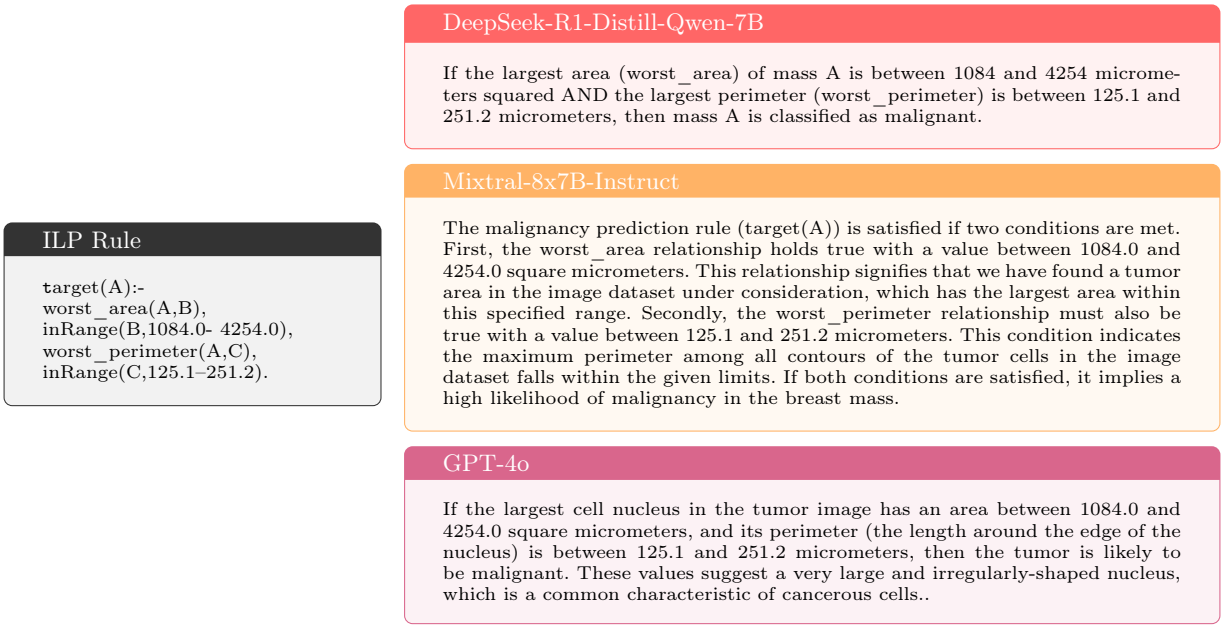
Explanation

Breakdown:
`worst_area(A, B)`: The largest area measured among all the cell nuclei in the sample image.
`inRange(B, 1084.0-4254.0)`: The area falls between 1084.0 and 4254.0 square micrometers.
`worst_perimeter(A, C)`: The largest perimeter (outline length) recorded for any nucleus.
`inRange(C, 125.1-251.2)`: The perimeter is between 125.1 and 251.2 micrometers.

Human-Readable Version:
If the largest cell nucleus in the tumor image has an area between 1084.0 and 4254.0 square micrometers, and its perimeter (the length around the edge of the nucleus) is between 125.1 and 251.2 micrometers, then the tumor is likely to be malignant. These values suggest a very large and irregularly shaped nucleus, which is a common characteristic of cancerous cells.

Implementation and Computational Performance The natural language translations were generated using three LLM architectures: GPT-4 (accessed via the official web interface), Mixtral-8x7B-Instruct, and DeepSeek-R1-Distill-Qwen-7B (both executed locally via LM Studio). All local inference was performed on an NVIDIA RTX 4090 laptop GPU to ensure hardware consistency. Benchmarking revealed significant performance differences between the locally-run models: DeepSeek-R1 demonstrated approximately $5\times$ faster inference speeds compared to Mixtral-8x7B under identical hardware conditions and equivalent parameter loading configurations. While GPT-4's cloud-based implementation provided stable outputs, its response times exhibited greater variability due to network latency and server-side queuing. All models processed the same set of logic rules in standardised prompt formats, with generation times systematically recorded. This performance advantage of the distilled DeepSeek model suggests its particular suitability for real-time applications where rapid translation of formal logic is prioritised.

Fig. 5: ILP rule with model-generated translations and clarity scores



Experimental Study To examine how different representations of learned logical rules affect human comprehension, we designed an experiment comparing formal symbolic outputs to their natural language equivalents. Motivated by ongoing interest in the interpretability of ILP-generated logical rules, our study focused on evaluating how effectively these rules and their translations by large language models (LLMs) convey meaning to human readers. Specifically, we aimed to investigate whether translating formal logic into natural language improves clarity, and which styles of translation are perceived as most understandable.

We also evaluated ILP performance using the same set of labeled examples under two configurations: pre-defined background knowledge from `PyGol` and BK generated by our hybrid LLM-assisted method. Results on a held-out validation set show that while Mixtral-generated BK enabled syntactically valid and structurally useful rule induction, it performed slightly lower in accuracy and F1-score compared to PyGol's pre-defined BK, primarily due to the additional complexity required for ensuring semantic precision (see Table 2).

Table 2: Comparison of Background Knowledge (BK) generated by PyGol and LLM (Language Model - Mixtral-8x7B-Instruct)

	Accuracy	**F1-Score**
PyGol generated (pre-defined) BK	0.947	0.937
LLM generated BK	0.916	0.894

By treating Mixtral as a knowledge constructor, we extend the role of LLMs from translation to induction scaffolding. This experiment not only tests the feasibility of LLMs in generating structured, interpretable BK, but also informs best practices for future applications of neural-symbolic integration. Our results suggest that while LLMs can support BK generation in a hybrid setup, fully replacing statistical routines with LLM-generated logic remains challenging due to variability, lack of numerical precision, and context limitations.

Participants and Procedure The interpretability experiments were conducted over a two-day period in May 2025 with 83 students from the University of Surrey, all of whom had prior exposure to formal logic or computational linguistics through coursework. Participants were presented with logic rules expressed

in two formats: a symbolic output generated by PyGol (ILP) and natural language translations produced by large language models (LLMs). The survey aimed to evaluate the comparative clarity and usability of these representations in conveying logical constructs. Of the 83 respondents, 77% (64 participants) found the translated natural language text more understandable, while 23% (19 participants) preferred the ILP output. Among the latter group, 53% rated the ILP output as *Clear* or *Very Clear*, though 5% described it as *Unclear* or worse. In contrast, the natural language translations were rated as *Very Clear* or *Clear* by 84% of respondents (54 out of 64), with none labeling them as *Unclear* or *Very Unclear*.

The survey was administered digitally via Microsoft Forms, with responses collected anonymously. To minimise bias, the order of ILP and natural language outputs was randomised. Participants were instructed to review all materials thoroughly, and the self-paced design allowed for deliberation without time constraints. For a summary of preference and clarity ratings, see Figure 6.

Limitations The condensed two-day window and fixed LLM comparison order may have introduced fatigue or priming effects. However, the homogeneity of the participant pool (logic-trained students) strengthens internal validity for assessing domain-specific comprehension.

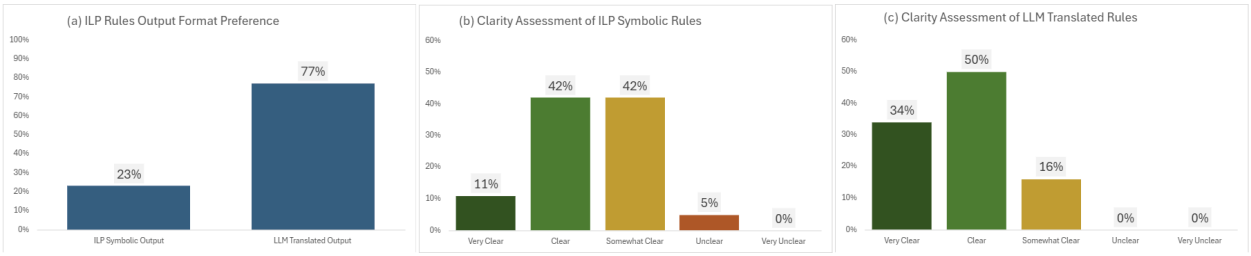


Fig. 6: Survey results: (a) ILP vs LLM Preference, (b) Clarity Assessment of ILP Output, (c) Clarity Assessment of LLM Translated Output

6 Discussion

This study explored the dual capacities of large language models (LLMs) within symbolic reasoning workflows. These roles included generating background knowledge (BK) and translating induced ILP rules into natural language. The results from these two experiments varied significantly, offering useful insights into the strengths and limitations of LLMs for these tasks.

In the BK generation task, LLMs such as Mixtral demonstrated that they can produce syntactically correct, Prolog-compatible representations from structured tabular data. However, when applied to precision-dependent tasks like numerical discretisation, their limitations became apparent. LLMs were unable to consistently model full distributions or maintain reproducible outputs across runs. While we attempted to mitigate these shortcomings by combining Python's statistical capabilities with LLM formatting, this introduced additional system complexity. Ultimately, while we were able to obtain reasonably acceptable results from LLM-based BK generation after overcoming the above challenges, the time, resources, and complexity involved in achieving those results were significant.

By contrast, the rule translation experiment demonstrated that LLMs hold significant promise in improving interpretability. We found that each model performed well in translating symbolic ILP rules into clear and accurate plain English explanations. Even users with some background in logic preferred the natural language versions. This indicates that LLMs can serve as effective tools for bridging the gap between formal symbolic reasoning and human understanding, which is especially important in real-world applications of explainable AI.

Taken together, these experiments highlight an important boundary in the integration of LLMs and ILP. While LLMs are well suited to enhancing accessibility and user interaction, they are currently less effective in

generating structured, precision-critical inputs required by logic-based systems. Future work may explore the use of LLMs to complement symbolic learners such as PyGol in areas beyond their intended design. While PyGol is primarily developed as an ILP learner focused on logic rule induction, it is not optimised for tasks requiring flexible abstraction or natural language interfacing. Leveraging LLMs in such areas could help extend the functionality of ILP systems and support more adaptable, interpretable neuro-symbolic frameworks.

7 Conclusion

This paper presents a hybrid framework that integrates LLMs into the ILP pipeline to enhance interpretability and automated generation of BK. In our experiments, we examined two key roles: translating symbolic ILP rules into natural language and generating background knowledge (BK) from raw tabular data.

Our findings show that LLMs can significantly improve the accessibility of logic-based outputs by producing fluent and comprehensible rule translations, as supported by human evaluation. However, their use for BK generation requires further investigation.

Together, these experiments highlight the value of LLMs as downstream collaborators in symbolic AI pipelines. Future work could explore domain-tuned models for translation fidelity and investigate scalable architectures that balance symbolic rigor with the expressive power of language models.

Acknowledgments. This research was supported by the time, compute resources, and collaborative infrastructure generously provided by the Azaries® Ltd. We thank all members of the research group for their contributions to different phases of the project-from system implementation to experimental validation and manuscript preparation. In particular, we acknowledge the efforts of our colleagues in designing the ILP pipelines, configuring the language models, and ensuring the reproducibility of results. Their expertise and collective commitment were instrumental in shaping the outcomes presented in this paper. We also thank Dr. Dany Varghese for their valuable guidance and technical support with the ILP system PyGol.

Disclosure of Interests. Azaries® Ltd supported this research through access to computational resources and collaborative infrastructure. While this affiliation may present a potential institutional interest in the outcomes of this study, all analyses, evaluations, and findings have been conducted independently and objectively. No commercial funding or external financial incentives were received for this work.

References

1. Afroozi Milani, G., Cyrus, D., Tamaddoni-Nezhad, A.: Towards One-Shot Learning for Text Classification using Inductive Logic Programming. In: Costantini, S. et al. (eds.) ICLP 2023. EPTCS, vol. 385, pp. 69–79 (2023). https://doi.org/10.4204/EPTCS.385.9
2. Besta, M., et al.: Affordable AI Assistants with Knowledge Graph of Thoughts. arXiv preprint, arXiv:2504.02670 (2025). https://arxiv.org/abs/2504.02670
3. Betz, G., Voigt, C., Richardson, K.: Critical Thinking for Language Models. In: Proceedings of IWCS 2021, pp. 63–75. ACL (2021)
4. Bhagavatula, C., et al.: Abductive Commonsense Reasoning. arXiv preprint, arXiv:1908.05739 (2020). https://arxiv.org/abs/1908.05739
5. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165 (2020). https://arxiv.org/abs/2005.14165
6. Chaghazardi, Z., Fallah, S., Tamaddoni-Nezhad, A.: Explainable and Trustworthy Traffic Sign Detection for Safe Autonomous Driving: An Inductive Logic Programming Approach. In: Costantini, S. et al. (eds.) ICLP 2023. EPTCS, vol. 385, pp. 201–212 (2023). https://doi.org/10.4204/EPTCS.385.21
7. Clark, P., Tafjord, O., Richardson, K.: Transformers as Soft Reasoners over Language. In: Proceedings of IJCAI-20, pp. 3882–3890 (2020). https://allenai.org/data/ruletaker
8. Cornelio, C., Thost, V.: Synthetic Datasets and Evaluation Tools for Inductive Neural Reasoning. In: LPNMR 2021, pp. 118–134. Springer, Cham (2021). https://github.com/IBM/RuDaS

9. Cropper, A., Tamaddoni-Nezhad, A., Muggleton, S.H.: Meta-Interpretive Learning of Data Transformation Programs. In: Proceedings of the 24th International Conference on Inductive Logic Programming (ILP 2015)
10. Cropper, A., Dumančić, S., Evans, R., Muggleton, S.H.: Inductive Logic Programming at 30. Machine Learning, 111, 147–172 (2022). https://doi.org/10.1007/s10994-021-06089-1
11. Debnath, A.K., Lopez de Compadre, R.L., Debnath, G., Shusterman, A.J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry* **34**, 786–797 (1991).
12. DeepSeek-AI: DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv preprint arXiv:2501.12948 (2025). https://arxiv.org/abs/2501.12948
13. Gandarela, J.P., Carvalho, D., Freitas, A.: Inductive Learning of Logical Theories with LLMs. arXiv preprint, arXiv:2408.16779v2 (2025). https://arxiv.org/abs/2408.16779
14. Gontier, N., Sinha, K., Reddy, S., Pal, C.: Measuring Systematic Generalization in Neural Proof Generation with Transformers. arXiv preprint, arXiv:2009.14786 (2020). https://arxiv.org/abs/2009.14786
15. Han, S., et al.: FOLIO: Natural Language Reasoning with First-Order Logic. arXiv preprint, arXiv:2209.00840v3 (2024). https://arxiv.org/abs/2209.00840
16. Ignatiev, A., Narodytska, N., Marques-Silva, J.: Abduction-Based Explanations for Machine Learning Models. In: Proceedings of the AAAI-19, pp. 1511–1519. AAAI Press (2019)
17. Liu, H., et al.: LogiQA 2.0—An Improved Dataset for Logical Reasoning in Natural Language Understanding. IEEE/ACM Trans. Audio Speech Lang. Process., 31, 2947–2959 (2023). https://doi.org/10.1109/TASLP.2023.3293046
18. Lu, Z., et al.: MathGenie: Generating Synthetic Data with Question Back-translation. arXiv preprint, arXiv:2402.16352v2 (2024). https://arxiv.org/abs/2402.16352
19. Mistral AI: Mixtral of Experts: Open-weight Mixture of Experts Language Model (2023). https://mistral.ai/news/mixtral-of-experts/
20. Morishita, T., Morio, G., Yamaguchi, A., Sogawa, Y.: Learning Deductive Reasoning from Synthetic Corpus Based on Formal Logic. In: Proceedings of ICML 2023, PMLR vol. 202. https://github.com/hitachi-nlp/FLD
21. Morishita, T., Morio, G., Yamaguchi, A., Sogawa, Y.: Enhancing Reasoning Capabilities of LLMs via Principled Synthetic Logic Corpus. In: Proceedings of NeurIPS 2024. https://arxiv.org/abs/2411.12498
22. Muggleton, S.: Inductive logic programming. New Generation Computing **8**(4), 295–318 (1991)
23. Muggleton, S., De Raedt, L.: Inductive logic programming: Theory and methods. Journal of Logic Programming **19/20**, 629–679 (1994)
24. Muggleton, S.H., Schmid, U., Zeller, C., Tamaddoni-Nezhad, A., Besold, T.R.: Ultra-Strong Machine Learning: Comprehensibility of Programs Learned with ILP. *Machine Learning*, **107**(7), 1119–1140 (2018). https://doi.org/10.1007/s10994-018-5707-3
25. OpenAI: GPT-4o Technical Report (2024). https://openai.com/index/gpt-4o
26. Polu, S., Sutskever, I.: Generative Language Modeling for Automated Theorem Proving. arXiv preprint, arXiv:2009.03393 (2020). https://arxiv.org/abs/2009.03393
27. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* **21**(140), 1–67 (2020). http://jmlr.org/papers/v21/20-074.html
28. Tafjord, O., Dalvi Mishra, B., Clark, P.: ProofWriter: Generating Implications, Proofs, and Abductive Statements over Natural Language. In: Findings of the ACL-IJCNLP 2021, pp. 3621–3634 (2021). https://aclanthology.org/2021.findings-acl.317
29. Tamaddoni-Nezhad, A., et al.: Human-Machine Scientific Discovery. In: Muggleton, S., Chater, N. (eds.) Human-Like Machine Intelligence, pp. 279–315. Oxford University Press (2021). https://doi.org/10.1093/oso/9780198862536.003.0015
30. Varghese, D., Milani, G.A., Tamaddoni-Nezhad, A.: Towards Enhancing LLMs with Logic-based Reasoning: A Meta Inverse Entailment Approach. Department of Computer Science, University of Surrey (2024). https://github.com/hmlr-lab/Compact-Language-Model
31. Wang, M., Deng, J.: Learning to Prove Theorems by Learning to Generate Theorems. In: NeurIPS, vol. 33, pp. 27894–27906 (2020). https://github.com/princeton-vl/MetaGen
32. Wei, J., et al.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In: NeurIPS, vol. 35, pp. 24824–24837 (2022)
33. Wolberg, W., Mangasarian, O., Street, N., Street, W.: Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository (1993). https://doi.org/10.24432/C5DW2B
34. Wolf, T., et al.: Transformers: State-of-the-Art Natural Language Processing. In: Proceedings of EMNLP 2020: System Demonstrations, pp. 38–45. https://arxiv.org/abs/1910.03771
35. Wu, Z., et al.: Reasoning or Reciting? Exploring LLMs Through Counterfactual Tasks. In: NAACL 2024, pp. 1819–1862 (2024)

36. Xu, F., et al.: Are Large Language Models Really Good Logical Reasoners? A Comprehensive Evaluation and Beyond. arXiv preprint, arXiv:2306.09841 (2024). https://arxiv.org/abs/2306.09841
37. Yang, Z., Dong, L., Du, X., Cheng, H., Cambria, E., Liu, X., Gao, J., Wei, F.: Language Models as Inductive Reasoners. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2024), Volume 1: Long Papers, pp. 209–225. Association for Computational Linguistics (2024). https://github.com/ZonglinY/Inductive_Reasoning
38. Yeo, W.J., et al.: How Interpretable are Reasoning Explanations from Prompting Large Language Models?. arXiv preprint, arXiv:2402.11863 (2024). https://arxiv.org/abs/2402.11863
39. Young, N., et al.: AbductionRules: Training Transformers to Explain Unexpected Inputs. In: Findings of ACL 2022, pp. 218–227 (2022)
40. Yuxuan Wan, Wenxuan Wang, Yiliu Yang, Youliang Yuan, Jen-tse Huang, Pinjia He, Wenxiang Jiao, and Michael Lyu. 2024. LogicAsker: Evaluating and Improving the Logical Reasoning Ability of Large Language Models. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 2124–2155, Miami, Florida, USA. Association for Computational Linguistics. https://github.com/yxwan123/LogicAsker
41. Zhang, Y., et al.: RuAG: Learned-Rule-Augmented Generation for Large Language Models. arXiv preprint, arXiv:2411.03349 (2024). https://arxiv.org/abs/2411.03349

A Appendix

A.1 Examples of Rule Translations by Language Models

This section contains examples of natural language translations of ILP rules produced by three different language models: GPT-4o, DeepSeek-R1-Distill-Qwen-7B, and Mixtral-8x7B-Instruct. Each translation corresponds to a symbolic rule generated by the PyGol ILP system.

Each rule set is grouped by model and listed with its original symbolic rule for comparison and evaluation. More examples are provided in the supplementary material.

GPT-4o - Original Rule (ILP):

```
target(A):-mean_perimeter(A,B),inRange(B,103.4--188.5),
          mean_texture(A,C),inRange(C,22.07--39.28)
```

Breakdown:

- `mean_perimeter(A,B)`: This is the average perimeter of the nuclei — essentially the average length around each nucleus.
- `inRange(B,103.4-188.5)`: Perimeter is between 103.4 and 188.5 micrometers.
- `mean_texture(A,C)`: This is the average texture of the nuclei, reflecting how consistent or varied the brightness levels are inside the nuclei (i.e., how smooth or rough they look).
- `inRange(C,22.07-39.28)`: Texture lies between 22.07 and 39.28.

Translation:

If the average perimeter of the cell nuclei falls between 103.4 and 188.5 micrometers, and the average texture (variation in brightness across each nucleus) is between 22.07 and 39.28, then the tumor is likely to be malignant. Large, irregular-shaped cells with uneven textures are typically seen in cancer, indicating that the cells are not growing in a uniform, healthy way.

DeepSeek-R1-Distill-Qwen-7B - Original Rule (ILP):

```
target(A):-mean_perimeter(A,B),inRange(B,103.4--188.5),
          mean_texture(A,C),inRange(C,22.07--39.28)
```

Translation:

If the mean perimeter (mean_perimeter) of mass A is between 103.4 and 188.5 micrometers AND the mean texture (mean_texture) is between 22.07 and 39.28, then mass A is classified as benign.

Mixtral-8x7B-Instruct - Original Rule (ILP):

```
target(A) :- mean_perimeter(A,B), inRange(B,103.4--188.5),
             mean_texture(A,C), inRange(C,22.07--39.28)
```

Translation:

The malignancy prediction rule (target(A)) is satisfied if two conditions are met. Firstly, the mean_perimeter relationship must be true with a value between 103.4 and 188.5 micrometers for the average perimeter of tumor cells in the image dataset. Secondly, the mean_texture relationship holds true if it has a value between 22.07 and 39.28, indicating the texture homogeneity among the tumor cells within the dataset. If both conditions are satisfied, it implies a high likelihood of malignancy in the breast mass.